



Open Source Used In talos_crawl-manager-go main-git-f0d1e27-115

Cisco Systems, Inc.

www.cisco.com

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

Text Part Number: 78EE117C99-2696284397

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

In your requests please include the following reference number 78EE117C99-2696284397

Contents

[1.1 lz4 1.9.3](#)

[1.1.1 Available under license](#)

[1.2 zlib 1.3.1](#)

[1.2.1 Available under license](#)

[1.3 zstd 1.3.1](#)

[1.3.1 Available under license](#)

[1.4 librdkafka 2.10.0-r0](#)

[1.4.1 Notifications](#)

[1.4.2 Available under license](#)

[1.5 openssl 3.0.15](#)

[1.5.1 Available under license](#)

[1.6 golang 1.25.0](#)

[1.6.1 Available under license](#)

[1.7 curl 8.14.1-r2](#)

[1.7.1 Available under license](#)

[1.8 zgo.at-zcache 2.4.1](#)

[1.8.1 Available under license](#)

[1.9 ncurses 6.5_p20241006-r3](#)

[1.9.1 Available under license](#)

[1.10 aws-sdk-go-v2-service-dynamodbstreams 1.32.0](#)

[1.10.1 Available under license](#)

[1.11 goautoneg 0.0.0-20191010083416-a7dc8b61c822](#)

[1.11.1 Available under license](#)

[1.12 smithy-go 1.23.1](#)

[1.12.1 Available under license](#)

[1.13 aws-sdk-go-v2-internal-v4a 1.4.11](#)

- 1.13.1 Available under license
- 1.14 golang-lru 2.0.7**
 - 1.14.1 Available under license
- 1.15 alpine-keys 2.5-r0**
 - 1.15.1 Available under license
- 1.16 shadow 4.16.0-r1**
 - 1.16.1 Available under license
- 1.17 x-text 0.30.0**
 - 1.17.1 Available under license
- 1.18 aws-sdk-go-v2-service-internal-endpoint-discovery 1.11.11**
 - 1.18.1 Available under license
- 1.19 perks 1.0.1**
 - 1.19.1 Available under license
- 1.20 aws-sdk-go-v2-service-internal-accept-encoding 1.13.2**
 - 1.20.1 Available under license
- 1.21 utmps 0.1.2.3-r2**
 - 1.21.1 Available under license
- 1.22 go-playground-locale 0.14.1**
 - 1.22.1 Available under license
- 1.23 validator 10.28.0**
 - 1.23.1 Available under license
- 1.24 libcap 2.71-r0**
 - 1.24.1 Available under license
- 1.25 logrus 1.9.3**
 - 1.25.1 Available under license
- 1.26 xxhash 2.3.0**
 - 1.26.1 Available under license
- 1.27 aws-sdk-go-v2-service-internal-presigned-url 1.13.11**
 - 1.27.1 Available under license
- 1.28 pam 1.6.1-r1**
 - 1.28.1 Available under license
- 1.29 ca-certificates 20250911-r0**
 - 1.29.1 Available under license
- 1.30 prometheus-client 1.23.2**
 - 1.30.1 Available under license
- 1.31 aws-sdk-go-v2-feature-ec2-imds 1.18.11**
 - 1.31.1 Available under license
- 1.32 aws-sdk-go-v2-internal-endpoints 2.7.11**
 - 1.32.1 Available under license

- 1.33 go-playground-universal-translator 0.18.1**
 - 1.33.1 Available under license
- 1.34 wojtekwanczyk-average 0.0.0-20240221125520-eb722bafaeb3**
 - 1.34.1 Available under license
- 1.35 apk-tools 2.14.6-r3**
 - 1.35.1 Available under license
- 1.36 aws-sdk-go-v2 1.39.4**
 - 1.36.1 Available under license
- 1.37 aws-sdk-go-v2-service-sts 1.38.9**
 - 1.37.1 Available under license
- 1.38 aws-sdk-go-v2-config 1.31.15**
 - 1.38.1 Available under license
- 1.39 x-crypto 0.43.0**
 - 1.39.1 Available under license
- 1.40 aws-sdk-go-v2-internal-ini 1.8.4**
 - 1.40.1 Available under license
- 1.41 protobuf-go 1.36.10**
 - 1.41.1 Available under license
- 1.42 readline 8.2.13-r0**
 - 1.42.1 Available under license
- 1.43 skalibs 2.14.3.0-r0**
 - 1.43.1 Available under license
- 1.44 kelseyhightower-envconfig 1.4.0**
 - 1.44.1 Available under license
- 1.45 bridgeutils 1.7.1-r2**
 - 1.45.1 Available under license
- 1.46 libbsd 0.12.2-r0**
 - 1.46.1 Available under license
- 1.47 aws-sdk-go-v2-service-sso 1.29.8**
 - 1.47.1 Available under license
- 1.48 google-uuid 1.6.0**
 - 1.48.1 Available under license
- 1.49 aws-sdk-go-v2-aws-protocol-eventstream 1.7.2**
 - 1.49.1 Available under license
- 1.50 go-urn 1.4.0**
 - 1.50.1 Available under license
- 1.51 go.yaml.in-yaml 2.4.3**
 - 1.51.1 Available under license
- 1.52 x-net 0.46.0**

- 1.52.1 Available under license
- 1.53 aws-sdk-go-v2-credentials 1.18.19**
 - 1.53.1 Available under license
- 1.54 aws-sdk-go-v2-service-dynamodb 1.52.2**
 - 1.54.1 Available under license
- 1.55 busybox 1.37.0-r13**
 - 1.55.1 Available under license
- 1.56 musl 1.2.5-r9**
 - 1.56.1 Available under license
- 1.57 pax-utils 1.3.8-r1**
 - 1.57.1 Available under license
- 1.58 openssl 3.3.5-r0**
 - 1.58.1 Available under license
- 1.59 alpine-base 3.21.5-r0**
 - 1.59.1 Available under license
- 1.60 procfs 0.19.1**
 - 1.60.1 Available under license
- 1.61 tre 0.8.0-r2**
 - 1.61.1 Available under license
- 1.62 aws-sdk-go-v2-internal-configsources 1.4.11**
 - 1.62.1 Available under license
- 1.63 mime-type 1.4.10**
 - 1.63.1 Available under license
- 1.64 prometheus-client-model 0.6.2**
 - 1.64.1 Available under license
- 1.65 libmd 1.1.0-r0**
 - 1.65.1 Available under license
- 1.66 prometheus-common 0.67.1**
 - 1.66.1 Available under license
- 1.67 aws-sdk-go-v2-service-internal-s3shared 1.19.11**
 - 1.67.1 Available under license
- 1.68 confluentinc-confluent-kafka-go 2.12.0**
 - 1.68.1 Available under license
- 1.69 aws-sdk-go-v2-service-s3 1.88.7**
 - 1.69.1 Available under license
- 1.70 aws-sdk-go-v2-feature-dynamodb-attributevalue 1.20.19**
 - 1.70.1 Available under license
- 1.71 aws-sdk-go-v2-feature-s3-manager 1.19.15**
 - 1.71.1 Available under license

1.72 alpine-baselayout 3.6.8-r1

1.72.1 Available under license

1.73 aws-sdk-go-v2-service-internal-checksum 1.9.2

1.73.1 Available under license

1.74 busybox 1.37.0-r14

1.74.1 Available under license

1.75 aws-sdk-go-v2-service-ssooidc 1.35.3

1.75.1 Available under license

1.76 bash 5.2.37-r0

1.76.1 Available under license

1.77 x-sys 0.37.0

1.77.1 Available under license

1.78 zlib 1.3.1-r2

1.78.1 Available under license

1.79 iputils 20240905-r0

1.79.1 Available under license

1.80 cJSON 1.7.19-r0

1.80.1 Notifications

1.80.2 Available under license

1.1 lz4 1.9.3

1.1.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

Preamble

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

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

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

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

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

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

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

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

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law:

that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

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

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

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

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

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

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

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

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

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

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

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

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

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

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

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

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

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

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For

example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

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

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

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

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Copyright (c) 2014, lpsantil
All rights reserved.

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

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

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

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

This repository uses 2 different licenses :

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

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

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

LZ4 Library

Copyright (c) 2011-2016, Yann Collet
All rights reserved.

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

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

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

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

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

Upstream-Name: liblz4

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

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

Files: *

Copyright: (C) 2011-2020 Yann Collet

License: GPL-2+

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

1.2 zlib 1.3.1

1.2.1 Available under license :

Copyright notice:

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

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

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

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

Jean-loup Gailly

jloup@gzip.org

Mark Adler

madler@alumni.caltech.edu

/* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.3.1, January 22nd, 2024

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

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

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

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

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

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

*/

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

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

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

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

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

```
/* zlib.c -- zlib functions common to reading and writing gzip files
* Copyright (C) 2004-2024 Mark Adler
* For conditions of distribution and use, see copyright notice in zlib.h
*/
```

1.3 zstd 1.3.1

1.3.1 Available under license :

BSD-style without advertising clause

1.4 librdkafka 2.10.0-r0

1.4.1 Notifications :

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

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

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

1.4.2 Available under license :

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

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

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

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

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

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

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

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

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

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

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

By:

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

Share and Enjoy! :-)

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

"

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

"

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

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all

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

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

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

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

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

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

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

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

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

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

as of the date such litigation is filed.

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

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

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

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

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

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

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

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

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

to in writing, shall any Contributor be

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

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

You agree to indemnify,

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

END OF TERMS AND CONDITIONS

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

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

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

* Copyright (C) 2013 Mark Adler

* Version 1.1 1 Aug 2013 Mark Adler

*/

/*

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

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

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

misrepresented as being the original
software.

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

Mark Adler

madler@alumni.caltech.edu

*/

For sys/queue.h:

* Copyright (c) 1991, 1993

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

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright

* notice, this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

* 4. Neither the name of the University nor the names of its contributors

* may be used to endorse or promote products derived from this software

* without specific prior written permission.

*

* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

* IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE

* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

* SUCH DAMAGE.

*

* @(#)queue.h 8.5 (Berkeley) 8/20/94

* \$FreeBSD\$

For cJSON.c and cJSON.h:

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

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

furnished to do so, subject to the following conditions:

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

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

LICENSE

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill

2023, Confluent Inc.

All rights reserved.

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

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

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

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

LICENSE.cjson

For cJSON.c and cJSON.h:

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

to do so, subject to the following conditions:

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

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

LICENSE.crc32c

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

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

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

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

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

misrepresented as being the original software.

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

Mark Adler

madler@alumni.caltech.edu

*/

LICENSE.fnv1a

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

Please

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

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

By:

chongo <Landon Curt Noll> /oo/

<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

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

The MIT License

(MIT)

Copyright (c) 2014 Coda Hale

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

copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

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

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

LICENSE.lz4

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

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

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

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

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

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

LICENSE.murmur2

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

MurMurHash2 Library

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

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

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

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

LICENSE.nanopb

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

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

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

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

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

documentation would be appreciated but is not required.

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

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

LICENSE.opentelemetry

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

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

and conversions to other media types.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

License. However, in accepting such obligations, You may act only

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

END OF TERMS AND CONDITIONS

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

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

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

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

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

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

LICENSE.pycrc

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

=====

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

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

furnished to do so, subject to the following conditions:

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

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

LICENSE.queue

For

sys/queue.h:

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

* \$FreeBSD\$

LICENSE.regexp

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

"

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

"

LICENSE.snappy

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

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

Copyright 2011 Intel Corporation All Rights Reserved.

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

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

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

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

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

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

LICENSE.tinychread

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

License

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

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

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

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

LICENSE.wingetopt

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

```
/*  
 * Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>  
 *  
 * Permission to use, copy, modify, and distribute this software for any  
 * purpose with or without fee is hereby granted, provided that the above  
 * copyright notice and this permission notice appear in all copies.  
 *  
 */
```

```

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
  ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*_
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice,
  this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/
#####
# LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h #
# originally retrieved from http://github.com/andikleen/snappy-c #
# git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #
#####

```

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

Copyright 2011 Intel Corporation All Rights Reserved.

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

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

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

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

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

librdkafka - Apache Kafka C driver library

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

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

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

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

ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

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

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

Upstream-Name: librdkafka

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

License: BSD-2-clause

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

.

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

.

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

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

Files: *

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

License: BSD-2-clause

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

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

License: MIT

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

permit persons to whom the Software is
furnished to do so, subject to the following conditions:

.

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

.

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

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

Copyright: 2005, Google Inc.

2011, Intel Corporation

License: BSD-3-clause

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

.

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

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

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

.

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

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

Copyright: 2012-2014, Yann Collet

License: BSD-2-clause

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

Copyright: n/a

License: public-domain

License: public-domain

The files tagged with this license contain the following paragraphs:

.

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

Files: debian/*

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

License: BSD-2-clause

BSD-style without advertising clause

Copyright (c) 2016-2019, Magnus Edenhill

All rights reserved.

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

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

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

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

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

For files in src/nanopb :

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

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

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

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

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

distribution.

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

MurMurHash2 Library

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

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

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

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

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

Upstream-Name: librdkafka

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

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

.

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

.

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

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

Files: *

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

License: BSD-2-clause

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

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

License: MIT

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

.

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

.

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

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

Copyright: 2005, Google Inc.

2011, Intel Corporation

License: BSD-3-clause

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

.

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

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

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

.

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

Files: debian/*

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

License: BSD-2-clause

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

The MIT License (MIT)

Copyright (c) 2014 Coda Hale

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

furnished to do so, subject to the following conditions:

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

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

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

License

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

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

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

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

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

=====

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

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

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

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

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

1.5 openssl 3.0.15

1.5.1 Available under license :

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

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

Terms of the Perl programming language system itself

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

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

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

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

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

Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

Preamble

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

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

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

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

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

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

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

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

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

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

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

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

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

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

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

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

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring

the other work under the scope of these terms.

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

- a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,
- b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,
- c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

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

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

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

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

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these

terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

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

program.

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

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

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

That's all there is to it!

--- The Artistic License 1.0 ---

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

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

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

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media

cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 - d) make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
 - a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
 - b) accompany the distribution with the machine-readable source of the Package with your modifications.
 - c) accompany any non-standard executables with their corresponding Standard

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

d) make other distribution arrangements with the Copyright Holder.

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

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

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

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

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

The End

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

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

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

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

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

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

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

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

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.6 golang 1.25.0

1.6.1 Available under license :

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

```
package copyright
```

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

```
var copyright = []byte("Copyright")
```

```

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

```

```

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

```

```

f, err := os.Open(path)
if err != nil {
    t.Error(err)
    return nil
}
defer f.Close()
n, err := f.Read(buf)
if err != nil && err != io.EOF {
    t.Error(err)
    return nil
}
b := buf[:n]
if bytes.Contains(b, copyright) {
    return nil
}
for _, ok := range permitted {
    if bytes.HasPrefix(b, ok) {
        return nil
    }
}

```

```

}
t.Errorf("%s: missing copyright notice", path)
return nil
})
}

```

Copyright 2009 The Go Authors.

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

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

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

The Go source code and supporting files in this directory are covered by the usual Go license (see `../..../LICENSE`).

When building with `GOEXPERIMENT=boringcrypto`, the following applies.

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

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

Contributors to BoringSSL are required to follow the CLA rules for Chromium:

<https://cla.developers.google.com/clas>

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

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

See below

for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

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

27287199

27287880

27287883

OpenSSL License

```
/* =====
* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following
disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
```

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

Original SSLeay License

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
```

- * Copyright remains Eric Young's, and as such any Copyright notices in
- * the code are not to be removed.
- * If this package is used in a product, Eric Young should be given attribution
- * as the author of
- the parts of the library used.
- * This can be in the form of a textual message at program startup or
- * in documentation (online or textual) provided with the package.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 3. All advertising materials mentioning features or use of this software
- * must display the following acknowledgement:
- * "This product includes cryptographic software written by
- * Eric Young (eay@cryptsoft.com)"
- * The word 'cryptographic' can be left out if the routines from the library
- * being used are
- not cryptographic related :-).
- * 4. If you include any Windows specific code (or a derivative thereof) from
- * the apps directory (application code) you must include an acknowledgement:
- * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
- *
- * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
- * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
- POSSIBILITY OF
- * SUCH DAMAGE.
- *
- * The licence and distribution terms for any publically available version or
- * derivative of this code cannot be changed. i.e. this code cannot simply be
- * copied and put under another distribution licence
- * [including the GNU Public Licence.]
- */

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.

*

* Permission to use, copy, modify, and/or distribute this software for any

* purpose with or without fee is hereby granted, provided that the above

* copyright notice and this permission notice appear in all copies.

*

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES

* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF

* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY

* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES

* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION

* OF CONTRACT, NEGLIGENCE

OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN

* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

Some files from Intel carry the following license:

Copyright (c) 2012, Intel Corporation

#

All rights reserved.

#

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are

met:

#

* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

#

* Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the

distribution.

#

* Neither the name of the Intel Corporation nor the names of its

contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

#

#

THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION

""AS IS"" AND ANY

EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.

<https://developers.google.com/open-source/cla/individual>
<https://developers.google.com/open-source/cla/corporate>

Names should be added to this file as:
Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyouun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
Copyright (c) 2009 The Go Authors. All rights reserved.

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

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

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

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

the Work and Derivative Works thereof.

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

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

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

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

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

meet the following conditions:

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

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

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

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

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

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

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

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

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

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

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

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

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

Copyright 2015 The Go Authors.

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

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

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

Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

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

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

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

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

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

1.7 curl 8.14.1-r2

1.7.1 Available under license :

Copyright (c) <year> <owner>.

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

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

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

COPYRIGHT AND PERMISSION NOTICE

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

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose

with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

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

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

Software without prior written authorization of the copyright holder.

BSD-4-Clause (University of California-Specific)

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

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

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

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

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above

copyright notice and this permission notice appear in all copies.

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

COPYRIGHT AND PERMISSION NOTICE

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

All rights reserved.

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

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

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

1.8 zgo.at-zcache 2.4.1

1.8.1 Available under license :

Copyright (c) 2012-2019 Patrick Mylund Nielsen and the go-cache contributors

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

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

all copies or substantial portions of the Software.

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

1.9 ncurses 6.5_p20241006-r3

1.9.1 Available under license :

Upstream source <https://invisible-island.net/ncurses/ncurses-examples.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2020,2021 Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

Licence: X11

Files: aclocal.m4 package

Copyright: 2003-2019,2020 by Thomas E. Dickey

Licence: X11

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

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

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

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

Files: install-sh

Copyright: 1994 X Consortium

Licence: X11

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

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

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

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

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent `make` implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8

Upstream source <https://invisible-island.net/ncurses/ncurses.html>

This package is used for testing builds of ncurses.

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2020,2021 by Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

Licence: X11

Files: aclocal.m4 package

Copyright: 1996-2020,2021 by Thomas E. Dickey

Licence: X11

Files: doc/html/NCURSES-Programming-HOWTO.html

Copyright: 2001 by Pradeep Padala

Licence: X11

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

furnished to do so, subject to the following conditions:

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

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

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

Files:

install-sh

Copyright: 1994 X Consortium

Licence: X11

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

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

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

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

Files: progs/tset.c ncurses/tinfo/read_termcap.c

Copyright: 1980,1991,1992,1993 The Regents of the University of California.

License: BSD

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

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

of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

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

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

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

-- vile: txtmode file-encoding=utf-8
Copyright 2018-2020,2021 Thomas E. Dickey
Copyright 1998-2017,2018 Free Software Foundation, Inc.

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

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

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

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

-- vile:txtmode fc=72
-- \$Id: COPYING,v 1.10 2021/01/01 09:54:30 tom Exp \$
Upstream source <https://invisible-island.net/ncurses/Ada95.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2020,2021 by Thomas E. Dickey
Copyright: 1998-2016,2017 Free Software Foundation, Inc.
Licence: X11

Files: aclocal.m4 package

Copyright: 2010-2020,2021 by Thomas E. Dickey

Licence: X11

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

The above copyright notice and this permission notice shall be included

in all copies or substantial portions of the Software.

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

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

Files: install-sh

Copyright: 1994 X Consortium

Licence: X11

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

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

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

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except
as contained in this notice, the name of the X Consortium shall not
be used in advertising or otherwise to promote the sale, use or other deal-
ings in this Software without prior written authorization from the X Consor-
tium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent
'make' implicit rules from creating a file called install from it
when there is no Makefile.

This script is compatible with the BSD install script, but was written
from scratch. It can only install one file at a time, a restriction
shared with many OS's install programs.

On Debian systems, the complete text of the GNU General
Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8
Upstream source <https://invisible-island.net/ncurses/ncurses-examples.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2020,2021 Thomas E. Dickey
Copyright: 1998-2016,2017 Free Software Foundation, Inc.
Licence: X11

Files: aclocal.m4 package
Copyright: 2003-2020,2021 by Thomas E. Dickey
Licence: X11

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

The above copyright notice and this permission notice shall be included

in all copies or substantial portions of the Software.

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

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

Files: install-sh
Copyright: 1994 X Consortium
Licence: X11

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

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

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

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

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent 'make' implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8

1.10 aws-sdk-go-v2-service-dynamodbstreams 1.32.0

1.10.1 Available under license :

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

```

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

package licensemanager

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

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

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

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata

```

```

return out, nil
}

type CreateLicenseInput struct {

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

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

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

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

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

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

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

    // Product name.
    //
    // This member is required.
    ProductName *string

```

```

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

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

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

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

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

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

// License status.
Status types.LicenseStatus

// License version.
Version *string

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

noSmithyDocumentSerde
}

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

```

```

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

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

```



```

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

```

```

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

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter

```

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

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

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

```

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

package licensemanager

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

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

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

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

type ListReceivedLicensesInput struct {

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

```

```

//
// - Beneficiary
Filters []types.Filter

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

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

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

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

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}

```



```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
}

```

```

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

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

DeleteLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

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

```

```

// Lists the licenses for your account.

```

```

func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

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

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

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

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

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

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

```

```

// Token for the next set of results.
NextToken *string

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

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {

```

```

    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

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

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata

```


ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

InterceptAfterDeserialization

OperationDeserializer

InterceptBeforeDeserialization

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
```

Directory or an Amazon Web Services Managed Active Directory.</p>"

```
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {

      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
```

```

    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },

```

```

"output": {

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
        "smithy.api#required": {}
      }
    },
    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    }
  }
}

```



```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
      "smithy.api#required": {}
    }
  }
},
{
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity

```

provider to deregister.</p>"

```
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity

```

```

provider.</p>"
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
}

```

```

        },
        "smithy.api#required": {}
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
        "ActiveDirectoryIdentityProvider": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProviderArn": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        }
    },
}

```

```

    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that

```

contains the user information for the product subscription.</p>"

```
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV4List": {
```

```

    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:[1,7]:|([0-9a-fA-F]{1,4}:[1,6]:[0-9a-fA-F]{1,4})$)"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",

```

```

        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],

```



```

        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                ],
            },
        ],
    },

```

true

```
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          }
        ]
      },
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  }
]
```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",

```



```

    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        },
        {
          "type": "tree"
        },
        {
          "conditions": [],
          "error": "DualStack is enabled but this partition does not support DualStack",
          "type": "error"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            }
        }
    ]
}

```

```

    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
}

```



```

    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"

```

```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
  },

```

```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>"
      }
    },
    "ServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>"
      }
    }
  }
}

```

```

resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {

```

```

        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
    },
    {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
    },
    {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            }
        ]
    }
}

```

```

        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        }
    ]
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results

```

to return from a single request.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
```

```

    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
    </p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
},

```



```

    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
      }
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
      "type": "structure",
      "members": {
        "UserAssociations": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#UserAssociations",
          "traits": {
            "smithy.api#documentation": "<p>The list of user associations for the specified resource.</p>"
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "UserAssociations": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UserAssociations",
        "traits": {
          "smithy.api#documentation": "<p>The list of user associations for the specified resource.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

"traits": {
  "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
}
},
"Filters": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
  "traits": {
    "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
  }
}
},
"traits": {
  "smithy.api#input": {}
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
    },
    "traits": {
        "smithy.api#output":
    {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of a product for this user.</p>",
                    "smithy.api#required": {}
                }
            },
            "ProductUserArn": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a

```

reference to the credentials provider that's used for RDS license server user administration.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {

  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

```



```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
      "Endpoint": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
    "type": "string",

```

```

    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings

```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

"uri": "/user/StartProductSubscription"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

```



```

    "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
  }
},
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

```

    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

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

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

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

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

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

```

```

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

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

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```



```

    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if

```

```

err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

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

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

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

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

```

```

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

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

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err

```

```

!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                  path: sdkbase
                  ref: ${ github.base_ref }

```

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

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

```



```

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

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

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

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

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

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

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.

```

```

ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

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

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

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

noSmithyDocumentSerde
}

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

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

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

```

```

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

```

```

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

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

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

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{  
    }  
}
```

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

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

type CheckoutBorrowLicenseInput struct {

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//

the request.

//

// This member is required.

ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)

// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].

//

// [RFC 7518 Digital Signature with RSASSA-PSS]: <https://tools.ietf.org/html/rfc7518#section-3.5>

//

// This member is required.

DigitalSignatureMethod types.DigitalSignatureMethod

```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

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

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
```



```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

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

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

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

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

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

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

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.

```

```

//
// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

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

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
}

```

```

if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

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

```

```

c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```

```

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

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

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

```

)

// Deletes the specified license.

```
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }
```

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

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

type DeleteLicenseInput struct {

// Amazon Resource Name (ARN) of the license.

//

// This member is required.

LicenseArn *string

// Current version of the license.

//

// This member is required.

SourceVersion

*string

noSmithyDocumentSerde

}

type DeleteLicenseOutput struct {

// Date when the license is deleted.

DeletionDate *string

// License status.

Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde


```

}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```

```

}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
}

```

```

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

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
}
}

```

```
    OperationName: "DeleteLicense",  
  }  
}  
CheckInLicense  
Initialize stack step  
  spanInitializeStart  
  InterceptExecution  
  RegisterServiceMetadata  
  legacyEndpointContextSetter  
  SetLogger  
  OperationInputValidation  
  spanInitializeEnd  
Serialize stack step  
  spanBuildRequestStart  
  setOperationInput  
  ResolveEndpoint  
  InterceptBeforeSerialization  
  OperationSerializer  
  InterceptAfterSerialization  
Build stack step  
  ClientRequestID  
  ComputeContentLength  
  SetCredentialSourceMiddleware  
  UserAgent  
  AddTimeOffsetMiddleware  
  RecursionDetection  
  spanBuildRequestEnd  
Finalize stack step  
  spanRetryLoop  
  InterceptBeforeRetryLoop  
  Retry  
  InterceptAttempt  
  RetryMetricsHeader  
  ResolveAuthScheme  
  GetIdentity  
  ResolveEndpointV2  
  disableHTTPS  
  ComputePayloadHash  
  setLegacyContextSigningOptions  
  InterceptBeforeSigning  
  Signing  
  InterceptAfterSigning  
Deserialize stack step  
  AddRawResponseToMetadata  
  ErrorCloseResponseBody  
  CloseResponseBody  
  ResponseErrorWrapper  
  RequestIDRetriever
```

InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

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

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

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

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

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

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

// Checks in the specified license. Check in a license when it is no longer in use.

```

func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

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

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

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

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

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}

```



```

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

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```

spanInitializeEnd
 Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {

```

    "type": "integer"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
    "type": "long"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      }
    }
  }
}

```

```

    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if\n    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      }
    },
    "InstanceType": {

```



```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The instance type of the resource.</p>"
    }
  },
  "AccountID": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see License Manager User Guide.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see License Manager User Guide.</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  }
}

```

```

    },
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
      }
    },
    "OsVersion": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
      }
    },
    "SubscriptionProviderCreateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
      }
    },
    "SubscriptionProviderUpdateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
      }
    },
    "DualSubscription": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
      }
    },
    "RegisteredWithSubscriptionProvider": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        }
      ]
    }
  }

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {

```

```

        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "Boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
]

```

```

    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ]
  }
],
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",

```

```

        "argv": [
            {
                "ref": "Region"
            }
        ]
    },
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        }
      ]
    }
  ],
  "supportsFIPS"
}
],
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        },
        {
          "ref": "DualStack"
        }
      ]
    }
  ],
  "supportsDualStack"
}
],
"rules": [
  {
    "conditions": [],
    "type": "endpoint"
  },
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
]

```



```

    ],
    "type": "tree"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
}
],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ]
    }
  ]
}

```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
}
],
"rules": [
  {

```

```

"conditions": [
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {

            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [

      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        }
    ],

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonaws.com.cn"
    }
  }
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {

```



```

        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
    
```

```

    "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
}
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists

```

the running Amazon EC2 instances that were discovered with commercial Linux\n subscriptions.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptionInstances"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Instances"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
<ul>\n
<li>\n
<p>\n
<code>AccountID</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>DualSubscription</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>InstanceID</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>InstanceType</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>ProductCode</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Region</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Status</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>UsageOperation</code>\n
</p>\n
</li>\n
</ul>\n
<p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
<ul>\n
<li>\n
<p>\n
<code>contains</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>equals</code>\n
</p>\n
</li>\n
<li>\n
<p>\n
<code>Notequal</code>\n
</p>\n
</li>\n
</ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    }
  },
  "NextToken": {
```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                }
            ],

```

```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
      "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
      returned results will include data aggregated across your accounts in\n    Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n        </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n                <code>contains</code>\n            </p>\n            </li>\n        <li>\n            <p>\n                <code>>equals</code>\n            </p>\n            </li>\n        <li>\n            <p>\n                <code>NotEqual</code>\n            </p>\n            </li>\n    </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    }
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListRegisteredSubscriptionProviders"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "RegisteredSubscriptionProviders"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\t\tin the list.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/RegisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
          "smithy.api#required": {}
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {

```



```

"type": "structure",
"members": {
  "SubscriptionProviderArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
    "traits": {
      "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderSource": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
    "traits": {
      "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\tRed Hat subscriptions.</p>"
    }
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",

```

```

        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
        }
    },
    "traits": {}
},

```



```

    "Type": {
      "target": "smithy.api#String",
      "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {

```

```

    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        }
      ]
    }
  }

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": true
        }
      }
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated"
    }
  }
}

```

```

from.</p>",
    "smithy.api#required": {}
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

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

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

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

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

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

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

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

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

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

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

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

1.11 goautoneg 0.0.0-20191010083416-a7dc8b61c822

1.11.1 Available under license :

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

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

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

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

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

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

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

1.12 smithy-go 1.23.1

1.12.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

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

met:

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

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

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

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

of such entity.

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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

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

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

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

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

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

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

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

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

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

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

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

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

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

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

1.13 aws-sdk-go-v2-internal-v4a 1.4.11

1.13.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step

```

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

package licensemanager

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

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

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,

```

```

c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

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

type CreateLicenseInput struct {

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

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

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

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

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

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

    // License name.
    //

```

```

// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

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

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

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

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

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

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

// License status.
Status types.LicenseStatus

// License version.
Version *string

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

noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
    if err

```



```

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

```

```

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

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

```

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

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

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

```

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

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

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

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

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

type ListReceivedLicensesInput struct {

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

```

```

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

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

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

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

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

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},

```

```

middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

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

```



```

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

```

```

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

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

```

DeleteLicense

```

```

Initialize stack step

```

```

spanInitializeStart

```

```

InterceptExecution

```

```

RegisterServiceMetadata

```

```

legacyEndpointContextSetter

```

```

SetLogger

```

```

OperationInputValidation

```

```

spanInitializeEnd

```

```

Serialize stack step

```

```
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licensemanager
```

```
import (
    "context"
```

```

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

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

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

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

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

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

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

    // Token for the next set of results.
    NextToken *string

```

```

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

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

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}

```

```

}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

DisassociateLicense
Initialize stack step
    spanInitializeStart
    InterceptExecution
    RegisterServiceMetadata
    legacyEndpointContextSetter
    SetLogger
    OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
    setOperationInput
    ResolveEndpoint
    InterceptBeforeSerialization
    OperationSerializer
    InterceptAfterSerialization
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
    RecursionDetection
    spanBuildRequestEnd
Finalize stack step
    spanRetryLoop
    InterceptBeforeRetryLoop
    Retry
    InterceptAttempt
    RetryMetricsHeader
    ResolveAuthScheme
    GetIdentity
    ResolveEndpointV2

```



```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    }
  },
  "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains

```

details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"

```
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
```

"smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"

```
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
```

```
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
    },
```

```
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
```

```
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
```

```
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
```

```

Directory.</p>",
    "smithy.api#length": {
        "min": 1,
        "max": 2
    }
},
"DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
},
"DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "SELF_MANAGED",
                "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-

```

```

zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      },
    },
  },

```

```

    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
  {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {

```



```

    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid"
      }
    }
  }
}

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^([a-z0-9-]{10})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",

```

```

    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
        "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    }
  },

```



```

    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    },
    "Products": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*"
      ]
    }
  }
}

```

```

        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",

```

```

        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    }
]

```



```

        ]
    }
],
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

"properties": {},
    "headers": {}
},
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    {
      "ref": "UseFIPS"
    },
    true
  ],
  "rules": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ],
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ],
            }
          ],
        }
      ],
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
},
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getattr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            },
            {
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    }
                ]
            }
        ]
    }
]

```

```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
},
],
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"

```

```

        },
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
    },

```

```

        "params": {
            "Region": "us-east-1",

"UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,

            "UseDualStack": true
        }
    }

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      }
    }
  ]
}

```



```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},

```

```

{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint":
{
      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```

```

    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    }
  }
}

```

```

    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>"
      }
    },
    "ServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  }
}

```

```

    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {

```

```

"type": "string",
"traits": {
  "smithy.api#enum": [
    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {

```

```

        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
    },
    {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {

```



```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single

```

```

request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/instance/ListInstances"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceSummaries"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {

```

```

    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
      "InstanceSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoints": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits":

```

```

{
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
}
},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```



```

        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
        }
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
    ]
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
        "method": "GET",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",

```

```

        "smithy.api#required": {}
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {

```

```

"type": "structure",
"members": {
  "RdsSalCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ]
  },

```

```

"traits": {
  "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/RegisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

```



```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {

        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",

```

```

        "smithy.api#length": {
            "min": 1
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services

```

Billing. For more information, see <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html> Viewing your monthly charges in the *Amazon Web Services Billing User Guide*.

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",

"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  }
}
```

```

        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        },
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
        ],
    },

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",

```



```

    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want

```

```

to tag.</p>",
    "smithy.api#httpLabel": {},
    "smithy.api#required": {}
  }
},
"Tags": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
      }
    },
    "UpdateSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
      "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",

```

```

        "traits": {
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    }
}

```

```

    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

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

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

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

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

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //

```

```

The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

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

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
}

```



```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

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

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

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

```

```

}

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

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

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

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);

```

```

err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {

```

```

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

```

```

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

```

on: [pull_request]

```

permissions:
  contents: read

```

```

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:

```

```

python-version: [3.9]

steps:
  - name: Checkout target
    uses: actions/checkout@v2
    with:
      path: sdkbase
      ref: ${ github.base_ref }
  - name: Checkout this ref
    uses: actions/checkout@v2
    with:
      path: new-ref
      fetch-depth: 0
  - name: Get Diff
    run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
  - name: Get Target Files
    run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
  - name: Checkout scancode
    uses: actions/checkout@v2
    with:
      repository: nexB/scancode-toolkit
      path: scancode-toolkit
      fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```



```

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

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

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

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

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

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

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

```

```

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

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

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

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

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

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

```

```

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}

```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```

```

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

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{

```

```

    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
  }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

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

```

```
// Checks out the specified license for offline use.
```

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

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

```

```

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

```

```
type CheckoutBorrowLicenseInput struct {
```

```

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

```

```

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

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

// License consumption token.
LicenseConsumptionToken *string

```

```

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

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

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}

```



```

}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

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

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

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

```

```

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

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

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

    noSmithyDocumentSerde
}

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

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

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

```

```

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

```

```

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

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "AssociateLicense",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

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

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

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

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

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

```



```

// License status.
Status types.LicenseDeletionStatus

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

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {

```

```

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

```

```

return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

```

CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

```

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

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

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

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

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

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

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

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

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

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

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

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

```

```

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

```



```

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

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}

```

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

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {

```

```

    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
    "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      }
    }
  }
}

```



```

    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```

```

    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",

```

```

"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    }
},
{
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
},
{
    "type": "tree"
}
],
}
],

```

```

    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"

```

```

        },
        true
    ]
}

],

"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        }
                    ]
                }
            ],
            "supportsFIPS"
        }
    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
},
],
"rules": [
{
    "conditions": [],
}
],
"rules": [
{

```



```

        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

```



```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "error": "DualStack is enabled but this partition does not support DualStack",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
}
],
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

```

```

    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  }
}

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```



```

    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
}

```

```

    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },

```

```

        {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Instances"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
        <ul>\n
          <li>\n
            <p>\n
              <code>AccountID</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>AmiID</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>DualSubscription</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>InstanceID</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>InstanceType</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>ProductCode</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>Region</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>Status</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>UsageOperation</code>\n
            </p>\n
          </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \t\define the behavior of the filter:</p>\n
        <ul>\n
          <li>\n
            <p>\n
              <code>contains</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>equals</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>Notequal</code>\n
            </p>\n
          </li>\n
        </ul>"
      }
    }
  },

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
    returned results will include data aggregated across your accounts in\n  Organizations.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptions"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Subscriptions"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
        that match one or\n  more sets of key-value pairs that you specify. For example, you can filter by the name of\n
        <code>Subscription</code> with an optional operator to see subscriptions that match,\n  partially match, or don't
        match a certain subscription's name.</p>\n  <p>The valid names for this filter are:</p>\n  <ul>\n
        <li>\n  <p>\n  <code>Subscription</code>\n  </p>\n  </li>\n  </ul>\n
        <p>The valid Operators for this filter are:</p>\n  <ul>\n  <li>\n  <p>\n  <code>contains</code>\n  </p>\n  </li>\n  <li>\n  <p>\n  <code>equals</code>\n  </p>\n  </li>\n  <li>\n  <p>\n  </li>\n  </ul>\n
      "
    }
  }
}

```

```

<code>Notequal</code>\n          </p>\n          </li>\n        </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",

```

```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSources": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
        }
      },
      "MaxResults": {

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {

```

```

"name": "Disabled",
  "value": "Disabled",
  "documentation": "Disabled OrganizationIntegration"
}
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {

```



```

subscription provider \n\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      }
    },
  },

```

```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",

```

```

        "documentation": "InProgress status"
    },
    {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [

```

```

        {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
    },

```


"smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
  "smithy.api#http": {
    "method": "DELETE",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",

  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tagKeys": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input":
    {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",

```

```

        "smithy.api#required":
    {}
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    }
}

```

```

    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

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

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

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

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

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

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

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

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensors and subsequently incorporated within the Work.

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

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

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

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

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

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

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

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

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

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

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

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

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

InterceptAfterDeserialization

OperationDeserializer

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

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.14 golang-lru 2.0.7

1.14.1 Available under license :

Copyright (c) 2014 HashiCorp, Inc.

Mozilla Public License, version 2.0

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of

version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's

modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any

Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code

Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any

liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4.

Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and

until such Contributor

explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section

2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward.

Except as provided in Section

10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source

Code Form that is

Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v.

2.0. If a copy of the MPL was not distributed with this file, You can obtain one at

<http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible
With Secondary Licenses", as defined by
the Mozilla Public License, v. 2.0.
This license applies to simplelrulist.go

Copyright (c) 2009 The Go Authors. All rights reserved.

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

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

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

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

// Copyright (c) HashiCorp, Inc.

// SPDX-License-Identifier: MPL-2.0

1.15 alpine-keys 2.5-r0

1.15.1 Available under license :

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the

rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

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

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

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=alpine-keys
pkgver=2.5
pkgrel=0
pkgdesc="Public keys for Alpine Linux packages"
url="https://alpinelinux.org"
# we install arch specific keys to /etc so we cannot do arch=noarch
arch="all"
license="MIT"
replaces="alpine-base"
```

1.16 shadow 4.16.0-r1

1.16.1 Available under license :

SPDX-License-Identifier: BSD-3-Clause

All files under this project either

1. fall under the BSD 3 clause license (by default).
2. carry an SPDX header declaring what license applies.

or

3. list a full custom license

This software is originally

* Copyright (c) 1989 - 1994, Julianne Frances Haugh

* All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions


```

* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the copyright holders or contributors may not be used to
*   endorse or promote products derived from this software without
*   specific
prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
* PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# Contributor: Patrycja Rosa <alpine@ptrcnll.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnll.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"
bulldir="$srcdir"
options="!check" # just headers

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

```

```

11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

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

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

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

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

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

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>

```

```

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

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

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

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

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

# Contributor: Dhruvin Gandhi <contact@dhruvin>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=py3-license-expression
_pkgname=license_expression
pkgver=30.4.0
pkgrel=0
pkgdesc="Library to parse, compare, simplify and normalize license expressions"
url="https://github.com/nexB/license-expression"
arch="noarch"
license="Apache-2.0"
depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"

```

```

source="https://files.pythonhosted.org/packages/source/l/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="${srcdir}/${_pkgname}-${pkgver}"

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

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

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

sha512sums="
cc7cd6b5bd8ea4c99b66f074871ffd7cc78b3e8b1b0d3afd9b95efdb7147c8293db2fac977d684fc34d3d60bd79066a06
dc2bff0a38b130d9d83c63538502db1
    license_expression-30.4.0.tar.gz
"

# License

All new source code committed to the shadow project is assumed to be made
available under the [BSD-3-Clause](../COPYING) license unless the submitter
specifies another license at that time. The shadow maintainers reserve the
right to refuse a submission if the license is deemed incompatible with the
goals of the project.

**Note**: old code may be made available under another license, check the
license tag for each file to get additional information.
# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-${pkgver}.tar.gz"

```

```

0001-bsd-games-scores-dir.patch
"
options="!check" # no tests

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

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

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

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>
# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>
pkgname=spdx-licenses
pkgver=3.24.0
pkgrel=0
pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

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

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

```

```

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

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

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

sha512sums="
6eb81424e1b3173dbd8e17859b2e15dfe1c2af8f4da59b7dad0729c038a4cbdba95c34c2c546285fcdbbd41562e39575e
6a6ebb9463b70121e0a12b99d1307e7  license-list-data-3.24.0.tar.gz
"

mini_sendmail - accept email on behalf of real sendmail

```

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

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

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

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

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

1.17 x-text 0.30.0

1.17.1 Available under license :

Copyright 2009 The Go Authors.

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

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

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

1.18 aws-sdk-go-v2-service-internal-endpoint-discovery 1.11.11

1.18.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution

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

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

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

// Creates a license.

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

```
    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,  
    c.addOperationCreateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CreateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

type CreateLicenseInput struct {

// License beneficiary.

//

// This member is required.

Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.

//

//

This member is required.

ClientToken *string

// Configuration for consumption of the license. Choose a provisional

// configuration for workloads running with continuous connectivity. Choose a

// borrow configuration for workloads with offline usage.

```

//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

```

```

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {

```

```

    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}

```

```

if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
    spanInitializeStart
    InterceptExecution
    RegisterServiceMetadata
    legacyEndpointContextSetter
    SetLogger
    OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
    setOperationInput
    ResolveEndpoint
    InterceptBeforeSerialization
    OperationSerializer
    InterceptAfterSerialization
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
    RecursionDetection
    spanBuildRequestEnd
Finalize stack step
    spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength


```

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns

```

```

...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type ListReceivedLicensesInput struct {

```

```

    // Filters to scope the results. The following filters are supported:

```

```

    //

```

```

    // - ProductSKU

```

```

    //

```

```

    // - Status

```

```

    //

```

```

    //

```

```

    // - Fingerprint

```

```

    //

```

```

    // - IssuerName

```

```

    //

```

```

    // - Beneficiary

```

```

    Filters []types.Filter

```

```

    // Amazon Resource Names (ARNs) of the licenses.

```

```

    LicenseArns []string

```

```

    // Maximum number of results to return in a single call.

```

```

    MaxResults *int32

```

```

    // Token for the next set of results.

```

```

    NextToken *string

```

```

    noSmithyDocumentSerde

```

```

}

```

```

type ListReceivedLicensesOutput struct {

```

```

    // Received license details.

```

```

    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }

```

```

}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return

```

```

err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

```

```

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```



```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```

```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput

```

ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```



```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```

```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"
          }
        }
      }
    }
  }
}

```

```

resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```



```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
    {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        }
    },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{

  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```

```

        "ref": "Region"
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [
                            {
                                "fn": "booleanEquals",
                                "argv": [
                                    {
                                        "ref": "UseDualStack"
                                    },
                                    true
                                ]
                            }
                        ],
                        "rules": [
                            {
                                "conditions": [
                                    {
                                        "fn": "booleanEquals",
                                        "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```

```

        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```



```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",

    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS":
false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint":
{
        "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS":
false,
      "UseDualStack": false
    }
  }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {

```

```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {

```



```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```

```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```

```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
  ],

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ]
    }
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```



```

"Product": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
    "smithy.api#required": {}
  }
},
"IdentityProvider": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"Status": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status of a product for this user.</p>",
    "smithy.api#required": {}
  }
},
"ProductUserArn": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
  }
},
"StatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
  }
},
"SubscriptionStartDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The start date of a subscription.</p>"
  }
},

```

```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {

```



```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        }
      },
      "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
      }
    },
    "UpdateSettings": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    }
},

"traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Removes the Grafana Enterprise license from a workspace.

```

func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
}
```



```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:

```

```
    region,  
    ServiceID:  ServiceID,  
    OperationName: "GetLicense",  
  }  
}
```

name: License Scan

on: [pull_request]

permissions:
 contents: read

jobs:

licensescan:

name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:

```
- name: Checkout target  
  uses: actions/checkout@v2  
  with:  
    path: sdkbase  
    ref: ${ github.base_ref }  
- name: Checkout this ref  
  uses: actions/checkout@v2  
  with:  
    path: new-ref  
    fetch-depth: 0  
- name: Get Diff  
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }  
${ github.sha } > refDiffFiles.txt  
- name: Get Target Files  
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt  
- name: Checkout scancode  
  uses: actions/checkout@v2  
  with:  
    repository: nexB/scancode-toolkit  
    path: scancode-toolkit  
    fetch-depth: 1  
- name: Set up Python  
  ${ matrix.python-version }  
  uses: actions/setup-python@v2  
  with:  
    python-version: ${ matrix.python-version }
```

```

# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

```

```

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}

```



```

}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licenssemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licenssemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }
}

```

```

result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,

```

```

c.addOperationCheckoutBorrowLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

```

```

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

```

```

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```



```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

```

```

// Amazon Resource Name (ARN) of the license.
//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

```

CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks in the specified license. Check in a license when it is no longer in use.
```

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
c.addOperationCheckInLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckInLicenseInput struct {
```

```
    // License consumption token.
```

```
    //
```

```

// This member is required.
LicenseConsumptionToken *string

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
  CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
  }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
  if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
  }
  err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
  if err != nil {
    return err
  }
  err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
  if err != nil {
    return err
  }
  if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
  }

  if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
  }
  if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
  }
  if err = addClientRequestID(stack); err != nil {
    return err
  }
  if
  err = addComputeContentLength(stack); err != nil {
    return err
  }
  if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
  }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```

```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}

```

```

}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt

```

```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```



```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    }
  },
  "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```

```

    "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  }
}

```



```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ]
  },

  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
}

```

```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```



```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

    },
    {
      "conditions": [],
      "error": "FIPS is enabled
but this partition does not support FIPS",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```



```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```

that you can use to filter the results by your specified criteria. \n \tFor example, you can specify

<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n

<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto streamline results:</p>\n

- \n <p>\n <code>AccountID</code>\n \n \n <p>\n <code>AmiID</code>\n \n \n <p>\n <code>DualSubscription</code>\n \n \n <p>\n <code>InstanceID</code>\n \n \n <p>\n <code>InstanceType</code>\n \n \n <p>\n <code>ProductCode</code>\n \n \n <p>\n <code>Region</code>\n \n \n <p>\n <code>Status</code>\n \n \n <p>\n <code>UsageOperation</code>\n

\n <p>For each filter, you can use one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n

- \n <p>\n <code>contains</code>\n \n \n <p>\n <code>equals</code>\n \n \n <p>\n <code>Notequal</code>\n

"

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {

```



```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\additional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ]
    }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```

```

    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",

```

```

        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```



```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required":
{}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.19 perks 1.0.1

1.19.1 Available under license :

Copyright (C) 2013 Blake Mizerany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.20 aws-sdk-go-v2-service-internal-accept-encoding 1.13.2

1.20.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

```

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

```

```
// License beneficiary.
//
// This member is required.
Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string
```

```

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
return err
}

```

```

}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

```

```

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart

```

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

```
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.

```

```

LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}

```



```

}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

```

```

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

```

```

}
}
DisassociateLicense
Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
  InterceptAfterDeserialization

```

```

OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    },
    "IsSharedActiveDirectory": {
      "target": "smithy.api#Boolean",

```



```

    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Ipv4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Ipv6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code> resource that contains information about the credential provider for user administration.</p>"
        }
      }
    }
  }
}

```

```

    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {

"target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {

```



```

        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
    }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
            }
        }
    },
    "traits": {

```

```

    "smithy.api#input": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},

"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
        "Subnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    }
  }
}

```

```

    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Products": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>A list of provided user-based subscription

```



```

products.</p>",
    "smithy.api#required": {}
  },
  "LastStatusCheckDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date of the last status check.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name

```

```

from the identity provider for the user.</p>",
    "smithy.api#required": {}
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },

```

```

    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}):|([0-9a-fA-F]{1,4}:{1,6}):[0-9a-fA-F]{1,4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",

```

```

    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
  },
  "UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
  },
  "UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
  },
  "Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
  }
},
"rules": [
  {

```

```

        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    },
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ],
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
            "type": "error"
        },
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ],
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
            "type": "error"
        },
        {
            "conditions": [],
            "endpoint": {
                "url": {
                    "ref": "Endpoint"
                }
            }
        }
    ]
}

```

```

    },

    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "isSet",
      "argv": [
        {
          "ref": "Region"
        }
      ]
    }
  ]
},
],
"rules": [
  {
    "conditions": [
      {
        "fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ]
      },
    ],
    "assign": "PartitionResult"
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          }
        ],
      },
    ],
  }
]
},
{

```

true


```

        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    },
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            }
        ]
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
],
"rules": [
    {

```

```

        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },

```

```

        "supportsFIPS"
    ]
},
true
]
}
],
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
            ],
            true
        }
    ]
},
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",

```

```

        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        },
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

```

```

        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

"UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,

```

```

        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

"endpoint": {
    "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
},
},
"params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,

```



```

        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"

```

```

        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license

```

```

server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
      },
      "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
      },
      "LicenseServerEndpointProvisioningStatus": {
        "target":

```

```

"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
  "traits": {
    "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",

```

```

        "value": "PROVISIONING_FAILED",
        "documentation": ""
    },
    {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    }
  },
  "NextToken": {

```



```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
    ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken"
      }
    }
  }
}

```

from a previously truncated response.</p>"

```
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListProductSubscriptions"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "ProductUserSummaries"
  }
}
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {

```

```

    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
        "smithy.api#http": {
          "method": "GET",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#readonly": {}
      }
    }
  }

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        },
        "traits": {
          "smithy.api#input": {}
        }
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
    },
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  }
},
  "traits": {
    "smithy.api#input": {}
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    }
  },

```

```

    "traits": {
      "smithy.api#output":
    },
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for this user.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for a

```

```

product for this user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
    }
  },
  "SubscriptionStartDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The start date of a subscription.</p>"
    }
  },
  "SubscriptionEndDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The end date of a subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {

```

```

    "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
  {
    "type": "structure",

```

```

"members": {
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
      "smithy.api#required": {}
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
  "type": "string",

```



```

    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      },
      "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
      "Endpoint": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",

```

```

    "input":
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
        }
    },
    "uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",

```

```

"members": {
  "Username": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",

```

```

        "uri": "/user/StopProductSubscription"
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "ProductUserArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",

```

```

    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {

```

```

    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation":
            "<p>The tags to apply to the specified resource.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```



```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}

      }
    },
    "TagKeys": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",

```

```

    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
    "IdentityProviderArn":
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
},
    "UpdateSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
        }
    }
},
    "traits": {
        "smithy.api#input": {}
    }
},
    "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
    "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
        "type": "structure",
        "members": {
            "AddSubnets": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
            "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
            "smithy.api#required": {}
        }
    },
    "RemoveSubnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
        }
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    }
},

"traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }

```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```



```

if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

// License version.
Version *string

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

// License details.
License *types.License

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}

```

```

}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:
        region,
    ServiceID:  ServiceID,
    OperationName: "GetLicense",
}
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                  path: sdkbase
                  ref: ${ github.base_ref }
            - name: Checkout this ref
              uses: actions/checkout@v2
              with:
                  path: new-ref
                  fetch-depth: 0
            - name: Get Diff
              run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }

```

```

${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Checks out the specified license.
```

```
//
```

```
// If the account that created the license is the same that is performing the
```

```
// check out, you must specify the account as the beneficiary.
```

```
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
```

```
(*CheckoutLicenseOutput, error) {
```

```
    if params == nil {
```

```

    params = &CheckoutLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckoutLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

```



```

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the checkout license.
    LicenseArn
    *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```

```

}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.

```

```

//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {

```

```

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}

```



```

}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}

```

```

}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

```

```

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
}

```

```

if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Deletes the specified license.

```

func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

```

```

}

result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*DeleteLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution

```

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks in the specified license. Check in a license when it is no longer in use.
```

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{}  
    }  
}
```

```
result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```

c.addOperationCheckInLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckInLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}

CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

```


InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
 "type": "integer"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
 "type": "long"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
 "type": "operation",

```

    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```



```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"

```

```

    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },

```



```
"OsVersion": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
},
"SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
},
"SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
},
"DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
    }
},
"RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tthird-party Linux subscription provider that you've registered with License Manager.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
},
```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
      }
    }
  }
}

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

{
  "conditions": [
    {
      "fn": "aws.partition",
      "argv": [
        {
          "ref": "Region"
        }
      ],
      "assign": "PartitionResult"
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        },
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {

```

```

        "ref": "PartitionResult"
    },
    "supportsFIPS"
]
}
]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
},
    "type": "tree"
},
{

```

```

        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],

```



```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

            "headers": {}
        },
        "type": "endpoint"
    }
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"conditions": [],
"error": "DualStack is enabled but this partition does not support DualStack",
"type": "error"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```



```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>NotEqual</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\t\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  },

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "InProgress",
                    "value": "InProgress",
                    "documentation": "InProgress status"
                },
                {
                    "name": "Completed",
                    "value": "Completed",
                    "documentation": "Completed status"
                },
                {
                    "name": "Successful",
                    "value": "Successful",
                    "documentation": "Successful status"
                },
                {
                    "name": "Failed",
                    "value": "Failed",
                    "documentation": "Failed status"
                }
            ]
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }

```

```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      },
    },
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {

```



```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.21 utmps 0.1.2.3-r2

1.21.1 Available under license :

Copyright (c) 2018-2024 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.22 go-playground-locales 0.14.1

1.22.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Go Playground

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.23 validator 10.28.0

1.23.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Dean Karn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.24 libcap 2.71-r0

1.24.1 Available under license :

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap/cap release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap/cap, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
- 3.

The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This
General Public License applies to most of the Free Software

Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities

other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide

a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such

an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),

conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL

ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which
makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap/psx release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap/psx, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3.
The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH

DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities

other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program

is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you;
rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to

apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL

ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey

the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which
makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause

- GPL v2.0

BSD 3-clause:

Redistribution and use in source and binary forms of libcap, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions.

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free

software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion

of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition,
mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING

ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```


The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public

License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

/* SPDX-License-Identifier: BSD-3-Clause OR LGPL-2.0-or-later */

Unless otherwise **explicitly** stated the following text describes the licensed conditions under which the contents of this module release may be distributed:

Redistribution and use in source and binary forms of this module, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

ALTERNATIVELY, this product may be distributed under the terms of the GNU Library General Public License, in which case the provisions of

the GNU LGPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU LGPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.25 logrus 1.9.3

1.25.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.26 xxhash 2.3.0

1.26.1 Available under license :

Copyright (c) 2016 Caleb Spare

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.27 aws-sdk-go-v2-service-internal-presigned-url 1.13.11

1.27.1 Available under license :

GetLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

```

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"

```

```

smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.

```

```

//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.

```

Status types.LicenseStatus

// License version.

Version *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

return err

}

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)

if err != nil {

return err

}

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},

middleware.After)

if err != nil {

return err

}

if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {

return fmt.Errorf("add protocol finalizers: %v", err)

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

return err

}

if err = addSetLoggerMiddleware(stack, options); err != nil {

return err

}

if err = addClientRequestID(stack); err != nil {

return err

}

if err = addComputeContentLength(stack); err != nil {

return err

}

if err = addResolveEndpointMiddleware(stack, options); err != nil {

return err

}

if err = addComputePayloadSHA256(stack); err != nil {

return err

}

if err = addRetry(stack, options); err != nil {

return err

```

}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return

```



```

err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
}

```

```

if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

```

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```

```

}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}

```



```

}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

```
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
```

```

    InterceptBeforeDeserialization
    AddTimeOffsetMiddleware
    RecordResponseTiming
    RequestResponseLogger
    InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters

```

```

[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
}

```

```

if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

DisassociateLicense
Initialize stack step
    spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
    setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware

```

```

RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {

```



```

    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    },
    "IsSharedActiveDirectory": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4

```

addresses that are used for the Active Directory.</p>",

```

    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  },
  "DomainIpv6List": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Ipv6List",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 2
      }
    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
    }
]
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {

```

"smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing. For more information, see Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    }
  },
}
```

```

    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active

```

Directory that contains user identity details.</p>",

```
    "smithy.api#required": {}
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
```

```

"members": {
  "SecretsManagerCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```



```

    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>  
resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ],
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Directory": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
      }
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {

        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",

```

```

"members": {
  "Username": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",

```

```

        "smithy.api#required": {}
    }
}
},

"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
        "Subnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
        {
            "Attribute": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
                }
            },
            "Operation": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
                }
            },
            "Value": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>Value of the filter.</p>"
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"Settings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

```

```

        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
    },

```



```

"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
    }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$|$))){4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}):{1,6}:[0-9a-fA-F]{1,4})$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
        }
      ]
    }
  }

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],

```

```

"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
  "UseDualStack": {
    "builtIn": "AWS::UseDualStack",

```

```

        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
]
],

```

```

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

{
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
    },
    "rules": [
        {
            "conditions": [

```



```

        {
            "fn": "aws.partition",
            "argv": [
                {
                    "ref": "Region"
                }
            ],
            "assign": "PartitionResult"
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        }
                    ],
                },
            ]
        },
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                }
                            ],
                            "supportsFIPS"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        ]
      }
    ]
  },

  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
]
}
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    {
      "ref": "UseFIPS"
    },
    true
  ]
},
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          },
          true
        ]
      }
    ],
    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  {
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",

```

```

        "type": "error"
    },
    {
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "endpoint": {
                    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
            }
        ]
    }
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
    },

```

```

        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },

```

```

    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",

"UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
    },

```

```

        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },

```



```

{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },

```

```

        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
}

```

```

    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    }
  },

```

```

    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license
server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The

```

```

<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {

```

```

    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",
          "value": "PROVISIONED",
          "documentation": ""
        },
        {
          "name": "DELETING",
          "value": "DELETING",
          "documentation": ""
        },
        {
          "name": "DELETION_FAILED",
          "value": "DELETION_FAILED",
          "documentation": ""
        },
        {
          "name": "DELETED",
          "value": "DELETED",
          "documentation": ""
        }
      ]
    }
  }
}

```

```

    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      },
      {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
},
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {

```



```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        }
    },

```

```

    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,

```

```

        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",
    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that

```

contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single

```

```

request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {

```



```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
        "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#readonly": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "Tags": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        }
    }
}

```

```

    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    }
  },

```

```

    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {

```

```

        "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {

```



```

    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
        "type": "union",

        "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
    {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }

},
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
    }
},
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
                }
            ]
        }
    }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",

"uri": "/user/StartProductSubscription"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {

```



```

        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "Tags": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {}
    }
}

```

```

        "min": 0,
        "max": 50
    },
    "smithy.api#sensitive": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",

```

```

"members": {
  "ResourceArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
      "smithy.api#httpLabel": {},
      "smithy.api#required": {}
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p></li> <li><p>Subnets which you want to remove the VPC endpoints from.</p></li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p></li> </ul>",
      "smithy.api#required": {}
    }
  }
}

```

```

    }

    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    }
}

```

```

    },

    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Removes the Grafana Enterprise license from a workspace.
```

```

func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```
    out := result.(*DisassociateLicenseOutput)
```

```

out.ResultMetadata = metadata
return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");

```

```

err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {

```



```

    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }

```

```

}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

```

on: [pull_request]

permissions:
contents: read

jobs:

licensescan:

name: License Scan
runs-on: ubuntu-latest
strategy:
matrix:
python-version: [3.9]

steps:

- name: Checkout target
uses: actions/checkout@v2
with:
path: sdkbase
ref: \${{ github.base_ref }}
- name: Checkout this ref
uses: actions/checkout@v2
with:
path: new-ref
fetch-depth: 0
- name: Get Diff
run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
\${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
uses: actions/checkout@v2
with:
repository: nexB/scancode-toolkit
path: scancode-toolkit
fetch-depth: 1
- name: Set up Python
\${{ matrix.python-version }}
uses: actions/setup-python@v2
with:
python-version: \${{ matrix.python-version }}

ScanCode

- name: Self-configure scancode
working-directory: ./scancode-toolkit
run: ./scancode --help
- name: Run Scan code on target
run: cat targetFiles.txt | while read filename; do echo ./sdkbase/\$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt

```

- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
  are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license.

```

```

//

```

```

// If the account that created the license is the same that is performing the

```

```

// check out, you must specify the account as the beneficiary.

```

```

func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
    c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type CheckoutLicenseInput struct {

```

```

//

```

```

Checkout type.

```

```

//

```

```

// This member is required.

```

```

CheckoutType types.CheckoutType

```



```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn

```

```

*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
        options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```

```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}

```

```

}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)

```

```

out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed

```

[]types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
```

```
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return
```

```
    err
```

```
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
```

```
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
```

```
    }
```

```
    if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```

return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
}

```



```

if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
    err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-

```

Grafana-Enterprise.html

```
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}
```

type AssociateLicenseInput struct {

// The type of license to associate with the workspace.

//

// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.

//

// This member is required.

LicenseType types.LicenseType

// The ID of the workspace to associate the license with.

//

// This member is required.

WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a

// Grafana Labs account. For more information, see [Link your account with Grafana Labs].

//

// [Link your account with Grafana Labs]: [https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-](https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise)
Grafana-Enterprise.html#AMG-workspace-register-enterprise

GrafanaToken *string

noSmithyDocumentSerde

}

type AssociateLicenseOutput

struct {

// A structure containing data about the workspace.

//

```

// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {

```

```

    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}

```

```

}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}

```



```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}

```

```

}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckInLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp    "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{}
```

```
    }  
  
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
    c.addOperationCheckInLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckInLicenseInput struct {
```

```
    // License consumption token.  
    //  
    // This member is required.  
    LicenseConsumptionToken *string
```

```
    // License beneficiary.  
    Beneficiary *string
```

```

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}

```

```

}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}

```



```

}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID:
        ServiceID,
    OperationName: "CheckInLicense",
}
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

```

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ]
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
      "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation":
                "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            }
        },
    },

```



```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated

```

```

data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",

```

```

        "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",

```

```

        "argv": [
            {
                "ref": "Endpoint"
            }
        ]
    },
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ]
        }
    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    }
}

```

```

        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [

"fn": "aws.partition",
            "argv": [
                {
                    "ref": "Region"
                }
            ],
            "assign": "PartitionResult"
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {

```

```

        "ref": "UseFIPS"
      },
      true
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "supportsFIPS"
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
]

```

```

        ]
    }
]
}
],
"rules": [
    {
        "conditions": [],

"rules": [

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {

"conditions": [

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ]
}

```



```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
      }
    ]
  }
}

```

```

        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],
                            "endpoint": {
                                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                                "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}

],
"type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"

```

```

    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {

```

```

        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```



```

{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
  "expect": {
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
  },
  "params": {

"Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
  "expect": {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "Missing region",

  "expect": {
    "error": "Invalid Configuration: Missing Region"
  }
}
],
"version": "1.0"

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",

```

```

    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
      <ul>\n
        <li>\n
          <p>\n
            <code>AccountID</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>AmiID</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>DualSubscription</code>\n
          </p>\n
          </li>\n
      </ul>
        "
        }
      }
    }
  }
}

```

```

</li>\n      <p>\n          <code>InstanceID</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>InstanceType</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>ProductCode</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>Region</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>Status</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>UsageOperation</code>\n      </p>\n  </li>\n</ul>\n<p>For each filter, you can use one of the following <code>Operator</code> values to \n \t\define the behavior of the filter:</p>\n  <ul>\n      <li>\n          <p>\n              <code>contains</code>\n          </p>\n      </li>\n      <li>\n          <p>\n              <code>>equals</code>\n          </p>\n      </li>\n      <li>\n          <p>\n              <code>Notequal</code>\n          </p>\n      </li>\n  </ul>\n  }\n},\n\"MaxResults\": {\n    \"target\": \"com.amazonaws.licenseanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n        \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n},\n\"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n        \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>\",\n        \"smithy.api#length\": {\n            \"min\": 1,\n            \"max\": 16384\n        }\n    }\n},\n\"traits\": {\n    \"smithy.api#documentation\":\n\"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases.\"\n},\n},\n\"com.amazonaws.licenseanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n        \"Instances\": {\n            \"target\": \"com.amazonaws.licenseanagerlinuxsubscriptions#InstanceList\",\n            \"traits\": {\n                \"smithy.api#documentation\": \"<p>An array that contains instance objects.</p>\"\n            }\n        }\n    },\n    \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n            \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \n\tfield

```

isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional

objects.

```
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
```

<p>Lists the Linux subscriptions that have been discovered. If you have linked your organization, the returned results will include data aggregated across your accounts in Organizations.</p>

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

    "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n                <code>contains</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>equals</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>NotEqual</code>\n            </p>\n        </li>\n    </ul>"
    }
},
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
},
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",

```

```
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
    }
  }
},
```

"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {

```
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
```

```
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
```

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
```

```
"type": "structure",
"members": {
  "SubscriptionProviderSources": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to return \n\t\t\tin the list.</p>"
}
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
    "smithy.api#range": {
      "min": 1,
      "max": 100
    }
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>"
  }
}
},

"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {
  "type": "structure",
  "members": {
    "RegisteredSubscriptionProviders": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList",
      "traits": {
        "smithy.api#documentation": "<p>The list of BYOL registration resources that fit the criteria \n\t\t\t\t\tyou specified in the request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with
```



```

    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "tags": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
      "smithy.api#http": {
        "code": 200,

```



```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,

```

```

\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-

```

```

\\.{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {

```



```

    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        },
        {
          "name": "PENDING",
          "value": "PENDING",
          "documentation": "PENDING status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":

```

```

{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```



```

        "target": "smithy.api#String"
    }
},
"traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input":
    {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },

```

```

    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":
      {}
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
      "smithy.api#required": {}
    }
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
    }
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated

```

```

from.</p>"
    }

    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.28 pam 1.6.1-r1

1.28.1 Available under license :

Unless otherwise *explicitly* stated the following text describes the licensed conditions under which the contents of this Linux-PAM release may be distributed:

Redistribution and use in source and binary forms of Linux-PAM, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License, in which case the provisions of the GNU

GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Unless otherwise *explicitly* stated the following text describes the licensed conditions under which the contents of this libpamc release may be distributed:

Redistribution and use in source and binary forms of libpamc, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of any author may not be used to endorse or promote products derived from this software without their specific prior written permission.

ALTERNATIVELY, this product may be distributed under the terms of the GNU Library General Public License (LGPL), in which case the provisions of the GNU LGPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU LGPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.29 ca-certificates 20250911-r0

1.29.1 Available under license :

```
* c_rehash.c - Create hash symlinks for certificates
* C implementation based on the original Perl and shell versions
*
* Copyright (c) 2013-2014 Timo Teras <timo.teras@iki.fi>
* All rights reserved.
*
* This software is licensed under the MIT License.
* Full license available at: http://opensource.org/licenses/MIT
*/
#
# This Source Code Form is subject to the terms of the Mozilla Public
# License, v. 2.0. If a copy of the MPL was not distributed with this
# file, You can obtain one at http://mozilla.org/MPL/2.0/.
Mozilla Public License
Version 2.0
1. Definitions
1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means
```


that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. “Executable Form”

means

any form of the work other than Source Code Form.

1.7. “Larger Work”

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. “License”

means this document.

1.9. “Licensable”

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. “Modifications”

means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. “Patent Claims” of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. “Secondary License”

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. “Source Code Form”

means the form of the work preferred for making modifications.

1.14. “You” (or “Your”)

means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for

any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in

Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s),

so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to

be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes

an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so

this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party’s ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - “Incompatible With Secondary Licenses” Notice

This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.

1.30 prometheus-client 1.23.2

1.30.1 Available under license :

Copyright (c) 2013 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are

met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Prometheus instrumentation library for Go applications

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

The following components are included in this product:

perks - a fork of <https://github.com/bmizerany/perks>

<https://github.com/beorn7/perks>

Copyright 2013-2015 Blake Mizerany, Bjrn Rabenstein

See <https://github.com/beorn7/perks/blob/master/README.md> for license details.

Go support for Protocol Buffers - Google's data interchange format

<http://github.com/golang/protobuf/>

Copyright 2010 The Go Authors

See source code for license details.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.31 aws-sdk-go-v2-feature-ec2-imds 1.18.11

1.31.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata

```

```

return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.
    //
    // This member is required.
    ProductName *string

```

```

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}

```



```

}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter

```

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step

spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

```

    InterceptBeforeDeserialization
    AddTimeOffsetMiddleware
    RecordResponseTiming
    RequestResponseLogger
    InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName

```

```

//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
}

```



```

if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

DeleteLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Lists the licenses for your account.

```

```

func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {

```

```

    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata

```

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

InterceptAfterDeserialization

OperationDeserializer

InterceptBeforeDeserialization

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
```


Directory or an Amazon Web Services Managed Active Directory.</p>"

```
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {

      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
            "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
    },
    "DomainNetworkSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
            "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                    "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
        },

```

```

"output": {

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
        "smithy.api#required": {}
      }
    },
    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    }
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```



```

    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
      "smithy.api#required": {}
    }
  }
},
{
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity

```

provider to deregister.</p>"

```
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}
```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity

```

```

provider.</p>"
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
}

```

```

        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
        "ActiveDirectoryIdentityProvider": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProviderArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        }
    },
}

```



```

    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that

```

contains the user information for the product subscription.</p>"

```
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV4List": {
```

```

    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:[1,7]:|([0-9a-fA-F]{1,4}:[1,6]:[0-9a-fA-F]{1,4})$)"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",

```

```

        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],

```

```

        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                ],
            },
        ],
    },

```


true

```
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          }
        ]
      },
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  }
]
```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",

```



```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    },
    {
      "type": "tree"
    },
    {
      "conditions": [],
      "error": "DualStack is enabled but this partition does not support DualStack",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "endpoint": {
    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            }
        }
    ]
}

```

```

    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
}

```

```

    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        }
    }

```



```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
  },

```

```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
      },
      "LicenseServerEndpointId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>"
        }
      }
    }
  }
}

```

```

resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {

```

```

        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
    },
    {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
    },
    {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            }
        ]
    }
}

```

```

        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```



```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results

```

to return from a single request.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
      }
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```



```

    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output":
  {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for this user.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a

```

reference to the credentials provider that's used for RDS license server user administration.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {

  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
},
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},

```



```

    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
      "Endpoint": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
    "type": "string",

```

```

    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings

```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

"uri": "/user/StartProductSubscription"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

```

```

    "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",

```

```

"members": {
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
  }
},
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

```

    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

```

```

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if

```

```

err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err

```

```

!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                  path: sdkbase
                  ref: ${{ github.base_ref }}

```

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.

```



```

ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

type CheckoutBorrowLicenseInput struct {

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//

the request.

//

// This member is required.

ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)

// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].

//

// [RFC 7518 Digital Signature with RSASSA-PSS]: <https://tools.ietf.org/html/rfc7518#section-3.5>

//

// This member is required.

DigitalSignatureMethod types.DigitalSignatureMethod

```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.

```

```

//
// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
}

```

```

if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,

```

```

c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

)

```
// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}
```

```
type DeleteLicenseInput struct {
```

```
    // Amazon Resource Name (ARN) of the license.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseArn *string
```

```
    // Current version of the license.
```

```
    //
```

```
    // This member is required.
```

```
    SourceVersion
```

```
    *string
```

```
    noSmithyDocumentSerde
```

```
}
```

```
type DeleteLicenseOutput struct {
```

```
    // Date when the license is deleted.
```

```
    DeletionDate *string
```

```
    // License status.
```

```
    Status types.LicenseDeletionStatus
```

```
    // Metadata pertaining to the operation's result.
```

```
    ResultMetadata middleware.Metadata
```

```
    noSmithyDocumentSerde
```

```

}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```



```

}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
}
}

```

```
    OperationName: "DeleteLicense",
  }
}
CheckInLicense
Initialize stack step
  spanInitializeStart
  InterceptExecution
  RegisterServiceMetadata
  legacyEndpointContextSetter
  SetLogger
  OperationInputValidation
  spanInitializeEnd
Serialize stack step
  spanBuildRequestStart
  setOperationInput
  ResolveEndpoint
  InterceptBeforeSerialization
  OperationSerializer
  InterceptAfterSerialization
Build stack step
  ClientRequestID
  ComputeContentLength
  SetCredentialSourceMiddleware
  UserAgent
  AddTimeOffsetMiddleware
  RecursionDetection
  spanBuildRequestEnd
Finalize stack step
  spanRetryLoop
  InterceptBeforeRetryLoop
  Retry
  InterceptAttempt
  RetryMetricsHeader
  ResolveAuthScheme
  GetIdentity
  ResolveEndpointV2
  disableHTTPS
  ComputePayloadHash
  setLegacyContextSigningOptions
  InterceptBeforeSigning
  Signing
  InterceptAfterSigning
Deserialize stack step
  AddRawResponseToMetadata
  ErrorCloseResponseBody
  CloseResponseBody
  ResponseErrorWrapper
  RequestIDRetriever
```

InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```

func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {

```

```

return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```


spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {

```

    "type": "integer"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
    "type": "long"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      }
    }
  }

```

```

    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if\n    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      }
    },
    "InstanceType": {

```

```

    "target": "smithy.api#String",
    "traits":
{
    "smithy.api#documentation": "<p>The instance type of the resource.</p>"
}
},
"AccountID": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
},
"Status": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
},
"Region": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
},
"UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n    <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n    operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
},
"ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n    operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
},
"LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
}

```



```

    },
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
      }
    },
    "OsVersion": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
      }
    },
    "SubscriptionProviderCreateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
      }
    },
    "SubscriptionProviderUpdateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
      }
    },
    "DualSubscription": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
      }
    },
    "RegisteredWithSubscriptionProvider": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        }
      ]
    }
  }

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {

```

```

        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "Boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
]

```

```

    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ]
  }
],
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",

```

```

        "argv": [
            {
                "ref": "Region"
            }
        ]
    },
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        }
      ]
    }
  ],
  "supportsFIPS"
}
]
},
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
}

```

```

    ],
    "type": "tree"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
}
],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ]
    }
  ]
}

```



```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
}
],
"rules": [
  {

```

```

"conditions": [
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        }
    ],

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonaws.com.cn"
    }
  }
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {

```

```

        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
    
```



```

    "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
}
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists

```

the running Amazon EC2 instances that were discovered with commercial Linux\n subscriptions.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptionInstances"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Instances"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n      <ul>\n        <li>\n          <p>\n            <code>AccountID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>AmiID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>DualSubscription</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>InstanceID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>InstanceType</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>ProductCode</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Region</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Status</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>UsageOperation</code>\n          </p>\n          </li>\n      </ul>\n      <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>equals</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Notequal</code>\n          </p>\n          </li>\n      </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            }
        ]
    }
}

```

```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
      "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
      returned results will include data aggregated across your accounts in\n    Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n                <code>contains</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>>equals</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>NotEqual</code>\n            </p>\n        </li>\n    </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    }
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListRegisteredSubscriptionProviders"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "RegisteredSubscriptionProviders"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\tin the list.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```



```
,
{
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
},
{
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
    "smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
    "smithy.api#http": {
        "method": "GET",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#readonly": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/RegisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
          "smithy.api#required": {}
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {

```



```

"type": "structure",
"members": {
  "SubscriptionProviderArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
    "traits": {
      "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderSource": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
    "traits": {
      "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
    }
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",

```

```

        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
        }
    },
    "traits": {}
},

```

```

    "Type": {
      "target": "smithy.api#String",
      "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {

```



```

    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        }
      ]
    }
  }

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": true
        }
      }
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated"
    }
  }
}

```

```

from.",
    "smithy.api#required": {}
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.32 aws-sdk-go-v2-internal-endpoints 2.7.11

1.32.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

```

ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of

```

```
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange
```



```

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err
= addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step

spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
"context"

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.

```



```

NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```

```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if

```

```

err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

```

```

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Lists the licenses for your account.
```

```

func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

}

out := result.(*ListLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {

```

```

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}

```



```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
DisassociateLicense
Initialize stack step
spanInitializeStart

```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    },
    "IsSharedActiveDirectory": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an

```

```

array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based

```

```

subscription.</p>",
    "smithy.api#required": {}
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

    "smithy.api#required": {}
    },
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)

```



```

license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
    "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
        }
    },
    "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
    }
},
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      }
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
        "type": "string",
        "traits": {

```

```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```



```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```

```

        "type": "endpoint"
    }
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
            "assign": "PartitionResult"
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```

```

        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            }
          ]
        }
      ],
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ]
    }
  ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",
  "type": "error"
}

],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```



```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {

```

```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```



```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results  
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken  
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```



```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```

```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```



```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
        "type": "structure",
        "members": {
            "IdentityProviderSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
          "smithy.api#length": {
            "min": 1
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
    "type": "string",
    "traits": {
      "smithy.api#length": {
        "min": 5,
        "max": 200
      }
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
},

```



```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
        "smithy.api#required": {}
    },
    "RemoveSubnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
        }
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}
```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```



```

"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID:  ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

```

```

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```



```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                path: sdkbase
                ref: ${ github.base_ref }
            - name: Checkout this ref
              uses: actions/checkout@v2
              with:
                path: new-ref
                fetch-depth: 0
            - name: Get Diff
              run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
                ${ github.sha } > refDiffFiles.txt
            - name: Get Target Files
              run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

```

```

- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the checkout license.
    LicenseArn
    *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}

```

```

}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"

```

```

smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

```



```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);

```

```

err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use  
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the  
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more  
// information, see [Upgrade a workspace to Grafana Enterprise].  
//
```

```
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
```

```
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))  
(*AssociateLicenseOutput, error) {  
    if params == nil {  
        params = &AssociateLicenseInput{}
```

```
    }  
  
    result, metadata, err := c.invokeOperation(ctx,  
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
  
    out := result.(*AssociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type AssociateLicenseInput struct {
```

```
    // The type of license to associate with the workspace.  
    //  
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free  
    // trials.  
    //  
    // This member is required.  
    LicenseType types.LicenseType
```

```
    // The ID of the workspace to associate the license with.
```

```

//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,

```

```

options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
}

```



```

if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Deletes the specified license.

```

```

func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {

```

```

return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp    "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks in the specified license. Check in a license when it is no longer in use.
```

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{}  
    }  
}
```

```
result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```

c.addOperationCheckInLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckInLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```



```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}

CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
 "type": "integer"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
 "type": "long"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
 "type": "operation",

```

    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
    },
  },
  "SubscriptionProviderSource": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

    "traits": {
      "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
    },
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
    },
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\t\tsuccessful subscription data request.</p>"
    },
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
    },
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\t\tfrom your registered third-party Linux subscription provider.</p>"
    },
  },
  "traits": {
    "smithy.api#output": {}
  },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",

```



```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"

```

```

    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
    "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
      }
    }
  }
}

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```



```

{
  "conditions": [
    {
      "fn": "aws.partition",
      "argv": [
        {
          "ref": "Region"
        }
      ],
      "assign": "PartitionResult"
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        },
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {

```

```

        "ref": "PartitionResult"
    },
    "supportsFIPS"
]
}
]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
},
    "type": "tree"
},
{

```

```

        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

            "headers": {}
        },
        "type": "endpoint"
    }
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"rules": [
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
          "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```



```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>Notequal</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```



```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\t\t\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  }
},

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }

```

```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
          "type": "string",
          "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home

```



```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.33 go-playground-universal-translator

0.18.1

1.33.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Go Playground

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.34 wojtekwanczyk-average 0.0.0-20240221125520-eb722bafae3

1.34.1 Available under license :

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.35 apk-tools 2.14.6-r3

1.35.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or

for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence

of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.

But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage

or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions.

You may

not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of

this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted

in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH

HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.36 aws-sdk-go-v2 1.39.4

1.36.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt

```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.

```

```

//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart

```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

```

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //

```

```

    - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {

```

```

return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}

```

```

}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

```

```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```



```

)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

```

```

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {

```

```

return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}

```

```

}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing

```

```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
    },
    "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
        "DomainName": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
            }
        },
        "DomainIpv4List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        },
        "DomainIpv6List": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 2
                }
            }
        }
    }
}

```

```

    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",

```



```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
},
"output": {

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":

```

```

{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the <code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for the endpoint. The settings include the type of license server and the Secrets Manager secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  }
},
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
            }
        },
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
        "SecretsManagerCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
      "smithy.api#required": {}
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
      "smithy.api#required": {}
    }
  }
},
{
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
      }
    }
  }
},
{
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        }
    }
},

```



```

    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      }
    ]
  },

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    }
  }
},

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {

```

```

        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
    {
        "type": "structure",
        "members": {
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
    },
    "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    }
  },

```



```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {

```

```

    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}):[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4}):([0-9a-fA-F]{1,4})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",

```

```

        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",

```

```

        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    },
    {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",

```

```

        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
    {
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {

```



```

        }
    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [

```



```

        {
            "fn": "getAttr",
            "argv": [

                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {

            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}

```

```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsDualStack"
                ]
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "endpoint": {

```

```

        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint":
{
    "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```



```

    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
                "traits": {
                    "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
                }
            },
            "StatusMessage": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
                }
            },
            "LicenseServerEndpointId": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
                }
            },
            "LicenseServerEndpointArn": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

        "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [

```

```

    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      }
    ]
  }
}

```

```

    },
    {
      "name": "NOT_APPLICABLE",
      "value": "NOT_APPLICABLE",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings  
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    }
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {

```



```

    "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

"type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't

```

empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"

```
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  }
}

```

```

},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```



```

        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

"traits": {
    "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
},
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        }
    }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output":
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
            "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    },
    "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {

```

```

    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"

            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
            "smithy.api#error": "client",
            "smithy.api#httpError": 404
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
        "type": "structure",
        "members": {
            "SecretId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
                    "smithy.api#length": {
                        "min": 1
                    }
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RDS_SAL",
            "value": "RDS_SAL",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
}

```



```

    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
},
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services
Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>\"",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",

"uri": "/user/StartProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},

```

```

        "smithy.api#required": {}
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",

```



```

        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
},
    "smithy.api#idempotent": {}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
          }
        },
        "UpdateSettings": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
          "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#required": {}
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {

```

```

        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.

```

```

WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
}

```

```

if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {

```

```

return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptExecution(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
return err
}
if err = addInterceptTransmit(stack, options); err != nil {
return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
}

```

```

if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)

```



```

    out.ResultMetadata = metadata
    return out, nil
}

```

```

type GetLicenseInput struct {

```

```

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

    // License version.
    Version *string

```

```

    noSmithyDocumentSerde
}

```

```

type

```

```

    GetLicenseOutput struct {

```

```

    // License details.
    License *types.License

```

```

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
}

```

```

if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}

```

name: License Scan

on: [pull_request]

permissions:
contents: read

jobs:
licensescan:
 name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:
 - name: Checkout target

```

    uses: actions/checkout@v2
    with:
      path: sdkbase
      ref: ${{ github.base_ref }}
- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"

```

```

"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

```

```

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
    if err = addClientUserAgent(stack, options); err != nil {

```



```

return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}

```

```

}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

```

```

}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5

```

```

//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

```

```

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);

```

```

err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}

```

```

    }
    return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```



```

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```

```

err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "AssociateLicense",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {

```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region

```



```

string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckInLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"
```

```

smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=

```

```

nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart

```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ],
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        },
        "Values": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
            }
        },
        "Operator": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
            "traits": {
                "smithy.api#documentation": "<p>An operator for filtering results.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":

```

```

"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

"LinuxSubscriptionsDiscovery": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
  "traits": {
    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
  }
},
"LinuxSubscriptionsDiscoverySettings": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
  "traits": {
    "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
  }
},
"Status": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
  }
},
"StatusMessage": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
  "traits": {
    "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
  }
},
"HomeRegions": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
  "traits": {
    "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
    }
  },
  "InstanceType": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The instance type of the resource.</p>"
    }
  },
  "AccountID": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {

```



```

    },
    "traits": {
      "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      },
      {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",

```

```

    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "String"
  },
  "UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "Boolean"
  },
  "UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
    "type": "Boolean"
  },
  "Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "String"
  }
},
"rules": [
  {
    "conditions": [
      {
        "fn": "isSet",
        "argv": [
          {
            "ref": "Endpoint"
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          }
        ]
      }
    ]
  }
]

```

```

        true
    ]
}

],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [

```

```

{
  "conditions": [
    {
      "fn": "isSet",
      "argv": [
        {
          "ref": "Region"
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseFIPS"
                },
                true
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              {
                "ref": "supportsDualStack"
              }
            ]
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],

    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

        },
        true

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

          "properties": {},

          "headers": {}
        },
        "type": "endpoint"
      ]
    },
    "type": "tree"
  ]
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
}
}

```

```

],
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}

```



```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "rules": [
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            }
        ],
        "type": "tree"
    }
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
],
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
}

```

```

    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
}

```

```

    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    disabled",
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

```

```

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
},

```



```

    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
      <ul>\n
        <li>\n
          <p>\n
            <code>AccountID</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>AmiID</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>DualSubscription</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>InstanceID</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>InstanceType</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>ProductCode</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>Region</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>Status</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>UsageOperation</code>\n
          </p>\n
          </li>\n
      </ul>\n
      <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \t\tdefine the behavior of the filter:</p>\n
      <ul>\n
        <li>\n
          <p>\n
            <code>contains</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>equals</code>\n
          </p>\n
          </li>\n
        <li>\n
          <p>\n
            <code>Notequal</code>\n
          </p>\n
          </li>\n
      </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      }
    }
  },

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
    returned results will include data aggregated across your accounts in\n    Organizations.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptions"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Subscriptions"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
        that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
        <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
        match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
        <li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
        <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
        <code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
        <code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
        <code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
        "type": "operation",
        "input": {
            "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
        },
        "output": {
            "target":

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
      "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "SecretArn": {

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
          }
        },
        "SubscriptionProviderStatus": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
          "traits": {
            "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
          }
        },
        "SubscriptionProviderStatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
          }
        },
        "LastSuccessfulDataRetrievalTime": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
          }
        }
      }
    }
  }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      }
    ],
  },

```

```

        {
            "name": "Successful",
            "value": "Successful",
            "documentation": "Successful status"
        },
        {
            "name": "Failed",
            "value": "Failed",
            "documentation": "Failed status"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
        }
    }
}

```

```

    }
  },
  "Type": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
    }
  },
  "InstanceCount": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
    "traits": {
      "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",

    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
          }
        ]
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
      "type":
    "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "ACTIVE",
            "value": "ACTIVE",
            "documentation": "ACTIVE status"
          },
          {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
          },
          {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
  {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
      }
    }
  }

```



```

        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
    "type": "map",
    "key":
{
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": true
        }
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {

```

```

    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
    "smithy.api#required": {}
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  }
}

```

```

    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step

spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

1.37 aws-sdk-go-v2-service-sts 1.38.9

1.37.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody

```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

```

```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

```

```

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}

```



```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization

```

Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint

InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd

```
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licenssemanager
```

```
import (
```

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

```

```

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {

```

```

    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}

```



```

}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if

```

```

err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

```

```

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Lists the licenses for your account.
```

```

func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

}

out := result.(*ListLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {

```

```

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
DisassociateLicense
Initialize stack step
spanInitializeStart

```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger


```

InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    },
    "IsSharedActiveDirectory": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an

```

```

array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based

```

```

subscription.</p>",
    "smithy.api#required": {}
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
    "smithy.api#error": "server"
  }
},

"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)

```

```

license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
{
    "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
            "smithy.api#required": {}
        }
    },
    "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
            "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
            "smithy.api#required": {}
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "InstanceId": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  }
}

```

```

    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
      },
      "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
        "type": "structure",
        "members": {
          "message": {
            "target": "smithy.api#String"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
          "smithy.api#error": "server",
          "smithy.api#httpError": 500
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
        "type": "list",
        "member": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
        }
      },
      "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
        "type": "string",
        "traits": {

```

```

    "smithy.api#pattern": "^[0-9a-fA-F]{1,4:}{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4:}{1,7}|([0-9a-fA-F]{1,4:}{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",

```

```

        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {

```

```

        "ref": "Endpoint"
    }
]

}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },

```



```

        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    }
                ]
            }
        ],
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},

```

```

        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]

```

```

        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
    },
    "params": {

```



```

        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS":
false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",

```

```

        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {

```

```

        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",

```

```

        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {

```



```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results  
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken  
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {

      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Product": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions

```



```

operation.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
  }
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
  "smithy.api#http": {
    "method": "GET",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#readonly": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {

```

```

    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
          }
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup": {
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    }
  }
}

```

```

    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":

```

```

{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
  }
},
"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }

},
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        },
        "traits": {
            "smithy.api#length": {
                "min": 0,
                "max": 50
            },
            "smithy.api#sensitive": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
        },

```



```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
  "smithy.api#http": {
    "method": "PUT",
    "uri": "/tags/{ResourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}

            }
        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}

            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      }
    }
  }
}

```

```

    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one

```

or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to activation servers.</p>",

```
        "smithy.api#required": {}
    },
    "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
        }
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}
```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
```

```

"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```



```

}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

```

```

noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                  path: sdkbase
                  ref: ${ github.base_ref }
            - name: Checkout this ref
              uses: actions/checkout@v2
              with:
                  path: new-ref
                  fetch-depth: 0
            - name: Get Diff
              run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
                  ${ github.sha } > refDiffFiles.txt
            - name: Get Target Files
              run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt

```

```

- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }
}

```



```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the checkout license.
    LicenseArn
    *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}

```

```

}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"

```

```

smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);

```



```

err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use  
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the  
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more  
// information, see [Upgrade a workspace to Grafana Enterprise].  
//
```

```
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
```

```
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))  
(*AssociateLicenseOutput, error) {  
    if params == nil {  
        params = &AssociateLicenseInput{}
```

```
    }  
  
    result, metadata, err := c.invokeOperation(ctx,  
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
  
    out := result.(*AssociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type AssociateLicenseInput struct {
```

```
    // The type of license to associate with the workspace.
```

```
    //
```

```
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free  
    // trials.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseType types.LicenseType
```

```
    // The ID of the workspace to associate the license with.
```

```

//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,

```

```

options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Deletes the specified license.

```

```

func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {

```



```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
return err
}
if err = addInterceptExecution(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks in the specified license. Check in a license when it is no longer in use.
```

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```

c.addOperationCheckInLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckInLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
}

```

```

if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
}

```



```

if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
 "type": "integer"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
 "type": "long"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
 "type": "operation",

```

    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
    },
  },
  "SubscriptionProviderSource": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

    "traits": {
      "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
    },
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
    },
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\t\tsuccessful subscription data request.</p>"
    },
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
    },
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\t\tfrom your registered third-party Linux subscription provider.</p>"
    },
  },
  "traits": {
    "smithy.api#output": {}
  },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",

```

```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"

```

```

    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {

```



```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
    "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
      }
    }
  }
}

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseFIPS"
                },
                true
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {

```

```

        "ref": "PartitionResult"
    },
    "supportsFIPS"
]
}
]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
},
    "type": "tree"
},
{

```



```

        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

            "headers": {}
        },
        "type": "endpoint"
    }
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"conditions": [],
"error": "DualStack is enabled but this partition does not support DualStack",
"type": "error"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",

```



```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>NotEqual</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

      "traits": {
        "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\tin the list.</p>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\t\t\tis the
nextToken from a previously truncated response.</p>"
      }
    }
  }
},

```



```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }
}

```



```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
          "type": "string",
          "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.38 aws-sdk-go-v2-config 1.31.15

1.38.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Creates a license.

```
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))  
(*CreateLicenseOutput, error) {  
    if params == nil {  
        params = &CreateLicenseInput{  
        }  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,  
    c.addOperationCreateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CreateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

type CreateLicenseInput struct {

// License beneficiary.

//

// This member is required.

Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.

//

//

This member is required.

ClientToken *string

// Configuration for consumption of the license. Choose a provisional

// configuration for workloads running with continuous connectivity. Choose a

// borrow configuration for workloads with offline usage.


```

//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

```

```

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {

```

```

    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}

```

```

if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
    spanInitializeStart
    InterceptExecution
    RegisterServiceMetadata
    legacyEndpointContextSetter
    SetLogger
    OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
    setOperationInput
    ResolveEndpoint
    InterceptBeforeSerialization
    OperationSerializer
    InterceptAfterSerialization
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
    RecursionDetection
    spanBuildRequestEnd
Finalize stack step
    spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength

```

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns

```



```

...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type ListReceivedLicensesInput struct {

```

```

    // Filters to scope the results. The following filters are supported:

```

```

    //

```

```

    // - ProductSKU

```

```

    //

```

```

    // - Status

```

```

    //

```

```

    //

```

```

    // - Fingerprint

```

```

    //

```

```

    // - IssuerName

```

```

    //

```

```

    // - Beneficiary

```

```

    Filters []types.Filter

```

```

    // Amazon Resource Names (ARNs) of the licenses.

```

```

    LicenseArns []string

```

```

    // Maximum number of results to return in a single call.

```

```

    MaxResults *int32

```

```

    // Token for the next set of results.

```

```

    NextToken *string

```

```

    noSmithyDocumentSerde

```

```

}

```

```

type ListReceivedLicensesOutput struct {

```

```

    // Received license details.

```

```

    Licenses []types.GrantedLicense

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }

```

```

}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return

```

```

err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

```

```

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

```

```

// Filters to scope the results. The following filters are supported:
//
// - Beneficiary
//
// - ProductSKU
//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {

```

```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {

```



```

    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput

```

ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {

```

```

    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    },
  },

```

```

"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",

```

```

    "smithy.api#required": {}
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current

```



```

state of the resource.</p>",
    "smithy.api#error": "server"
  }

},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
          "smithy.api#required": {}
        }
      },
      "LicenseServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>"
          }
        }
      }
    }
  }
}

```

```

resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ],
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License
Manager user-based subscriptions.</p>",
        "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
            }
        },
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
      }
    }
  },
  "traits": {

```



```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
        "type": "structure",
        "members":
    {
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        }
    },

```

```

"Operation": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information

```

about an identity provider.</p>",

```
    "smithy.api#required": {}
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
```

```
    "smithy.api#required": {}
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }
  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
```

```

    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "InstanceId": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
          }
        },
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details

```

```

about the identity provider.</p>",
    "smithy.api#required": {}
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$|$))){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {

```

```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
  },

```



```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

```

```

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
},
"UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
},
"UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
    "type": "boolean"
},
"Endpoint": {
    "builtIn": "SDK::Endpoint",
    "required": false,
    "documentation": "Override the endpoint used to send this request",
    "type": "string"
}
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",

```

```

        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    },
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [

{
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    },
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {

```

```

        "ref": "Region"
      }
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
        "fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "booleanEquals",
                "argv": [
                  {
                    "ref": "UseDualStack"
                  },
                  true
                ]
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [

```

```

        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        }
    ],
    },

    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
]

    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or

```

both",

```
        "type": "error"
      }
    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            }
          }
        ]
      }
    ]
  }
}
```

```

        "type": "endpoint"
    }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [

```



```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      },
      "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        },
        "params": {
          "Region": "us-east-1",
          "UseFIPS": false,
          "UseDualStack": true
        }
      },
      {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
          "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
          }
        }
      }
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
}

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS":
false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint":
{
        "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS":
false,
      "UseDualStack": false
    }
  }
}

```

```

    },
    {
      "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
      },
      "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {

```

```

        "url": "https://example.com"
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {

```

```

    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {

```

```

        "smithy.api#documentation": "<p>The type of license server.</p>"
    },
    "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",

```

```

        "documentation": ""
    },
    {
        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "ServerSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based

```

```

subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/ListIdentityProviders"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "IdentityProviderSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>An array of IdentityProviderSummary resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
      "smithy.api#required": {}
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ]
    }
  }

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {

```

```

"type": "structure",

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
  ]
}

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",

```

```

        "smithy.api#required": {}
    },
    "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
    },
    "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    }
},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {

```

```

        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
            }
        ]
    }
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      }
    },
  },

```

```

"Product": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
    "smithy.api#required": {}
  }
},
"IdentityProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"Status": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status of a product for this user.</p>",
    "smithy.api#required": {}
  }
},
"ProductUserArn": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
  }
},
"StatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
  }
},
"SubscriptionStartDate": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The start date of a subscription.</p>"
  }
},

```



```

    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        }
      ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }

  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {

```

```

    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#Settings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Product": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    }
}

```



```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
          }
        }
      },
      "IdentityProvider":

```

```

{
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    }
},
"Product": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
},
"ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
},
"Domain": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#StringList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
      "type": "list",
      "member": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "IdentityProviderArn":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
      }
    },
    "UpdateSettings": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    }
},

"traits": {
    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Removes the Grafana Enterprise license from a workspace.

```

func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {

```

```

    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if

```

```

err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:

```

```
    region,  
    ServiceID:  ServiceID,  
    OperationName: "GetLicense",  
  }  
}
```

name: License Scan

on: [pull_request]

permissions:
 contents: read

jobs:

licensescan:

name: License Scan
 runs-on: ubuntu-latest
 strategy:
 matrix:
 python-version: [3.9]

steps:

```
- name: Checkout target  
  uses: actions/checkout@v2  
  with:  
    path: sdkbase  
    ref: ${ github.base_ref }  
- name: Checkout this ref  
  uses: actions/checkout@v2  
  with:  
    path: new-ref  
    fetch-depth: 0  
- name: Get Diff  
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }  
${ github.sha } > refDiffFiles.txt  
- name: Get Target Files  
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt  
- name: Checkout scancode  
  uses: actions/checkout@v2  
  with:  
    repository: nexB/scancode-toolkit  
    path: scancode-toolkit  
    fetch-depth: 1  
- name: Set up Python  
  ${ matrix.python-version }  
  uses: actions/setup-python@v2  
  with:  
    python-version: ${ matrix.python-version }
```

```

# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

```

```

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}

```

```

}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licenssemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licenssemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,

```



```

c.addOperationCheckoutBorrowLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

```

```

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

```

```

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

```

```

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```



```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

```

```

// Amazon Resource Name (ARN) of the license.
//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}

```

```

}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {

```

```

    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

```

CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID

```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks in the specified license. Check in a license when it is no longer in use.
```

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
c.addOperationCheckInLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckInLicenseInput struct {
```

```
    // License consumption token.
```

```
    //
```



```

// This member is required.
LicenseConsumptionToken *string

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
  CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
  }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
  if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
  }
  err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
  if err != nil {
    return err
  }
  err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
  if err != nil {
    return err
  }
  if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
  }

  if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
  }
  if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
  }
  if err = addClientRequestID(stack); err != nil {
    return err
  }
  if
  err = addComputeContentLength(stack); err != nil {
    return err
  }
  if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
  }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```

```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}

```

```

}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt

```

```

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ]
    }
  }
}

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Values": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n  Filters can be used to match a set of resources by specific criteria.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    }
  }
}

```

```

    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {

```



```

"type": "service",
"version": "2018-05-10",
"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {

```

```

"additionalAllowedHeaders": [
  "*",
  "content-type",
  "x-amz-content-sha256",
  "x-amz-user-agent",
  "x-amzn-platform-id",
  "x-amzn-trace-id"
],
"additionalExposedHeaders": [
  "x-amzn-errortype",
  "x-amzn-requestid",
  "x-amzn-trace-id"
]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ]
      }
    ],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ]
  },

  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
}

```



```

    },
    {
      "conditions": [],
      "endpoint": {
        "url": {
          "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ]
    }
  ],
  "rules": [

```



```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
    },
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

        {
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    }
]

```

```

    },
    {
      "conditions": [],
      "error": "FIPS is enabled
but this partition does not support FIPS",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "rules": [

```

```

{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
},
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",

```



```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

```

```

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>An array of structures

```

that you can use to filter the results by your specified criteria. \n \tFor example, you can specify

<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n

<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto streamline results:</p>\n

- \n <p>\n <code>AccountID</code>\n \n \n <p>\n <code>AmiID</code>\n \n \n <p>\n <code>DualSubscription</code>\n \n \n <p>\n <code>InstanceID</code>\n \n \n <p>\n <code>InstanceType</code>\n \n \n <p>\n <code>ProductCode</code>\n \n \n <p>\n <code>Region</code>\n \n \n <p>\n <code>Status</code>\n \n \n <p>\n <code>UsageOperation</code>\n

<p>For each filter, you can use one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n

- \n <p>\n <code>contains</code>\n \n \n <p>\n <code>equals</code>\n \n \n <p>\n <code>Notequal</code>\n

"

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {

```

```

    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t organization, the
returned results will include data aggregated across your accounts in\n\t Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
<code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
<code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "Subscriptions": {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
    "traits": {
      "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the next request to retrieve \n\additional objects.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ]
    }
},

```

```

        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "Enabled",
                    "value": "Enabled",
                    "documentation": "Enabled OrganizationIntegration"
                },
                {
                    "name": "Disabled",
                    "value": "Disabled",
                    "documentation": "Disabled OrganizationIntegration"
                }
            ]
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
        "type": "list",
        "member": {
            "target": "smithy.api#String"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ]
    }
}

```



```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
  }
},
"SubscriptionProviderArn": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\tthat you requested.</p>"
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  }
}

```



```

    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {

```

```

        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
        "smithy.api#http": {
            "method": "PUT",
            "uri": "/tags/{resourceArn}"
        },
    },

```



```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
        }
    },
    "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input":
{}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ]
}

```



```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required":
{}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {

```

```

        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

}

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.39 x-crypto 0.43.0

1.39.1 Available under license :

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.40 aws-sdk-go-v2-internal-ini 1.8.4

1.40.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata

```

```

return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.
    //
    // This member is required.
    ProductName *string

```

```

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}

```

```

}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter

```


SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step

spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

```

    InterceptBeforeDeserialization
    AddTimeOffsetMiddleware
    RecordResponseTiming
    RequestResponseLogger
    InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName

```

```

//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}

```

```

}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

DeleteLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID


```

ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Lists the licenses for your account.

```

```

func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

```

```

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {

```

```

    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata

```

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

InterceptAfterDeserialization

OperationDeserializer

InterceptBeforeDeserialization

AddTimeOffsetMiddleware

RecordResponseTiming

RequestResponseLogger

InterceptTransmit

```
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
```

Directory or an Amazon Web Services Managed Active Directory.</p>"

```
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {

      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    }
  },
  "DomainCredentialsProvider": {
```



```

    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },

```

```

"output": {

"target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current  
state of the resource.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
        "smithy.api#required": {}
      }
    },
    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    }
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
      "smithy.api#required": {}
    }
  }
},
{
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity

```

provider to deregister.</p>"

```
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}
```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/DisassociateUser"
        },
        "smithy.api#idempotent": {}
    }
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity

```

```

provider.</p>"
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
}

```

```

        },
        "smithy.api#required": {}
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
        "ActiveDirectoryIdentityProvider": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "Settings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
                    "smithy.api#required": {}
                }
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
                    "smithy.api#required": {}
                }
            },
            "Status": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The status of the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProviderArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        }
    },
}

```

```

    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {

```



```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
            "smithy.api#required": {}
        }
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that

```

contains the user information for the product subscription.</p>"

```
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV4List": {
```

```

    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:[1,7]:|([0-9a-fA-F]{1,4}:[1,6]:[0-9a-fA-F]{1,4})$)"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",

```

```

        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],

```

```

        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                ],
            },
        ],
    },

```

true

```
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          }
        ]
      },
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  }
]
```



```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",

```



```

    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        },
        {
          "type": "tree"
        },
        {
          "conditions": [],
          "error": "DualStack is enabled but this partition does not support DualStack",
          "type": "error"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            }
        }
    ]
}

```

```

    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
}

```

```

    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        }
    }

```

```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```



```

    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
  },

```

```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>"
      }
    },
    "ServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>"
      }
    }
  }
}

```

```

resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {

```

```

        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
    },
    {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
    },
    {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            }
        ]
    }
}

```

```

        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {

```



```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results

```

to return from a single request.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
      }
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    }
  }
}

```



```

    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output":
  {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for this user.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a

```

reference to the credentials provider that's used for RDS license server user administration.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {

  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
  {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
          "smithy.api#required": {}
        }
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
      "Endpoint": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
    "type": "string",

```



```

    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings

```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

"uri": "/user/StartProductSubscription"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

```

```

"min": 0,
  "max": 50
},
"smithy.api#sensitive": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
    }
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
  }
},
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

```

    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

```

```

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if

```



```

err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err

```

```

!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                  path: sdkbase
                  ref: ${{ github.base_ref }}

```

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.

```

```

ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```



```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

type CheckoutBorrowLicenseInput struct {

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//

the request.

//

// This member is required.

ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)

// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].

//

// [RFC 7518 Digital Signature with RSASSA-PSS]: <https://tools.ietf.org/html/rfc7518#section-3.5>

//

// This member is required.

DigitalSignatureMethod types.DigitalSignatureMethod

```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```



```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.

```

```

//
// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
}

```

```

if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,

```

```

c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

)

```
// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}
```

```
type DeleteLicenseInput struct {
```

```
    // Amazon Resource Name (ARN) of the license.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseArn *string
```

```
    // Current version of the license.
```

```
    //
```

```
    // This member is required.
```

```
    SourceVersion
```

```
    *string
```

```
    noSmithyDocumentSerde
```

```
}
```

```
type DeleteLicenseOutput struct {
```

```
    // Date when the license is deleted.
```

```
    DeletionDate *string
```

```
    // License status.
```

```
    Status types.LicenseDeletionStatus
```

```
    // Metadata pertaining to the operation's result.
```

```
    ResultMetadata middleware.Metadata
```

```
    noSmithyDocumentSerde
```

```

}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```

```

}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
}

```



```

if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
}
}

```

```
    OperationName: "DeleteLicense",  
  }  
}  
CheckInLicense  
Initialize stack step  
  spanInitializeStart  
  InterceptExecution  
  RegisterServiceMetadata  
  legacyEndpointContextSetter  
  SetLogger  
  OperationInputValidation  
  spanInitializeEnd  
Serialize stack step  
  spanBuildRequestStart  
  setOperationInput  
  ResolveEndpoint  
  InterceptBeforeSerialization  
  OperationSerializer  
  InterceptAfterSerialization  
Build stack step  
  ClientRequestID  
  ComputeContentLength  
  SetCredentialSourceMiddleware  
  UserAgent  
  AddTimeOffsetMiddleware  
  RecursionDetection  
  spanBuildRequestEnd  
Finalize stack step  
  spanRetryLoop  
  InterceptBeforeRetryLoop  
  Retry  
  InterceptAttempt  
  RetryMetricsHeader  
  ResolveAuthScheme  
  GetIdentity  
  ResolveEndpointV2  
  disableHTTPS  
  ComputePayloadHash  
  setLegacyContextSigningOptions  
  InterceptBeforeSigning  
  Signing  
  InterceptAfterSigning  
Deserialize stack step  
  AddRawResponseToMetadata  
  ErrorCloseResponseBody  
  CloseResponseBody  
  ResponseErrorWrapper  
  RequestIDRetriever
```

InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp    "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```

func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {

```

```

return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {


```

    "type": "integer"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
    "type": "long"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      }
    }
  }

```

```

    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if\n    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      }
    },
    "InstanceType": {

```

```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The instance type of the resource.</p>"
    }
  },
  "AccountID": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see License Manager User Guide.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see License Manager User Guide.</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance details.</p>"
    }
  }
}

```

```

    },
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
      }
    },
    "OsVersion": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
      }
    },
    "SubscriptionProviderCreateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
      }
    },
    "SubscriptionProviderUpdateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
      }
    },
    "DualSubscription": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
      }
    },
    "RegisteredWithSubscriptionProvider": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    }
  }

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        }
      ]
    }
  }

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      }
    },
    "UseDualStack": {

```

```

        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "Boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
]

```

```

    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ]
  }
],
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",

```

```

        "argv": [
            {
                "ref": "Region"
            }
        ]
    },
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        }
      ]
    }
  ],
  "supportsFIPS"
]
},
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
},
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
]

```

```

    ],
    "type": "tree"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
}
],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ]
    }
  ]
}

```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
}
],
"rules": [
  {

```



```

"conditions": [
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {

            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [

      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        }
    ],

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
  }
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {

```

```

        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
    
```

```

    "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
}
},

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists

```

the running Amazon EC2 instances that were discovered with commercial Linux\n subscriptions.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptionInstances"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Instances"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n      <ul>\n        <li>\n          <p>\n            <code>AccountID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>AmiID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>DualSubscription</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>InstanceID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>InstanceType</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>ProductCode</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Region</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Status</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>UsageOperation</code>\n          </p>\n          </li>\n      </ul>\n      <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>equals</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Notequal</code>\n          </p>\n          </li>\n      </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            }
        ]
    }
}

```

```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
      "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
      returned results will include data aggregated across your accounts in\n    Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
          that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
          <code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
          match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
          <li>\n    <p>\n    <code>Subscription</code>\n    </p>\n    </li>\n    </ul>\n
          <p>The valid Operators for this filter are:</p>\n    <ul>\n    <li>\n    <p>\n
          <code>contains</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
          <code>equals</code>\n    </p>\n    </li>\n    <li>\n    <p>\n
          <code>Notequal</code>\n    </p>\n    </li>\n    </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    }
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListRegisteredSubscriptionProviders"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "RegisteredSubscriptionProviders"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\t\tin the list.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/RegisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
          "smithy.api#required": {}
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {

```



```

"type": "structure",
"members": {
  "SubscriptionProviderArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
    "traits": {
      "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderSource": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
    "traits": {
      "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
    }
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",

```

```

        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
        }
    },
    "traits": {}
},

```

```

    "Type": {
      "target": "smithy.api#String",
      "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {

```

```

    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        }
      ]
    }
  }

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": true
        }
      }
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated"
    }
  }
}

```

```

from.",
    "smithy.api#required": {}
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.41 protobuf-go 1.36.10

1.41.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.42 readline 8.2.13-r0

1.42.1 Available under license :

@c The GNU Free Documentation License.

@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,

@c hence no sectioning command or @node.

@display

Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

@end display

@enumerate 0

@item

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document @dfn{free} in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain @sc{ascii} without markup, Texinfo input format, LaTeX input format, @acronym{SGML} or @acronym{XML} using a publicly available @acronym{DTD}, and standard-conforming simple @acronym{HTML}, PostScript or @acronym{PDF} designed for human modification. Examples of transparent image formats include @acronym{PNG}, @acronym{XCF} and @acronym{JPG}. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, @acronym{SGML} or

@acronym{XML} for which the @acronym{DTD} and/or processing tools are not generally available, and the machine-generated @acronym{HTML}, PostScript or @acronym{PDF} produced by some word processors for output purposes only.

The ``Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher" means any person or entity that distributes copies of the Document to the public.

A section ``Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements", ``Dedications", ``Endorsements", or ``History".) To ``Preserve the Title" of such a section when you modify the Document means that it remains a section ``Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy

of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as

given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements" or ``Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements". Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements" or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,
you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a

standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History" in the various original documents, forming one section Entitled ``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this

License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract

a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See @uref{<http://www.gnu.org/copyleft/>}.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ``or any later version'' applies to it, you have the option of following the terms and conditions either of that specified version or

of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

RELICENSING

``Massive Multiauthor Collaboration Site" (or ``MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A

``Massive Multiauthor Collaboration" (or ``MMC")

contained in the

site means any set of copyrightable works thus published on the MMC site.

``CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and

license notices just after the title page:

```
@smallexample
```

```
@group
```

```
Copyright (C) @var{year} @var{your name}.
```

```
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled ``GNU
Free Documentation License".
```

```
@end group
```

```
@end smallexample
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ }Texts." line with this:

```
@smallexample
```

```
@group
```

```
with the Invariant Sections being @var{list their titles}, with
the Front-Cover
Texts being @var{list}, and with the Back-Cover Texts
being @var{list}.
```

```
@end group
```

```
@end smallexample
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

```
@c Local Variables:
```

```
@c ispell-local-pdict: "ispell-dict"
```

```
@c End:
```

```
GNU GENERAL PUBLIC LICENSE
```

```
Version 2, June 1991
```

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
```

```
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
```

```
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

```
Preamble
```


The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and

modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole

or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further

restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to
freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive

or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees"

and

"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that

Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of

your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section

7. This requirement modifies the requirement in section 4 to

"keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical

medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option

remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating

where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting

any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest

to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

1.43 skalibs 2.14.3.0-r0

1.43.1 Available under license :

Copyright (c) 2011-2024 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.44 kelseyhightower-envconfig 1.4.0

1.44.1 Available under license :

Copyright (c) 2013 Kelsey Hightower

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.45 bridgeutils 1.7.1-r2

1.45.1 Available under license :

```
/*
 * Copyright (C) 2000 Lennert Buytenhek
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License as
 * published by the Free Software Foundation; either version 2 of the
 * License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but
 * WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
```

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the

Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering

access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to

apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License,

or

(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.46 libbsd 0.12.2-r0

1.46.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:
*

Copyright:
Copyright 2004-2024 Guillem Jover <guillem@hadrons.org>
License: BSD-3-clause

Files:
include/bsd/err.h
include/bsd/stdlib.h
include/bsd/sys/param.h
include/bsd/unistd.h
src/bsd_getopt.c
src/err.c
src/fgetln.c
src/progname.c

Copyright:
Copyright 2005, 2008-2012, 2019 Guillem Jover <guillem@hadrons.org>
Copyright 2005 Hector Garcia Alvarez
Copyright 2005 Aurelien Jarno
Copyright 2006 Robert Millan
Copyright 2018 Facebook, Inc.
License: BSD-3-clause

Files:
include/bsd/netinet/ip_icmp.h
include/bsd/sys/bitstring.h
include/bsd/sys/queue.h
include/bsd/sys/time.h
include/bsd/timeconv.h
include/bsd/vis.h
man/bitstring.3bsd
man/errc.3bsd
man/explicit_bzero.3bsd
man/fgetln.3bsd
man/fgetwln.3bsd
man/fpurge.3bsd
man/funopen.3bsd
man/getbsize.3bsd
man/heapsort.3bsd
man/nlist.3bsd
man/pwcache.3bsd
man/queue.3bsd
man/radixsort.3bsd
man/reallocarray.3bsd
man/reallocf.3bsd
man/setmode.3bsd
man/strmode.3bsd
man/strnstr.3bsd
man/strtoi.3bsd
man/strtou.3bsd

man/unvis.3bsd
man/vis.3bsd
man/wcslcpy.3bsd
src/getbsize.c
src/heapsort.c
src/merge.c
src/nlist.c
src/pwcache.c
src/radixsort.c
src/setmode.c
src/strmode.c
src/strnstr.c
src/strtoi.c
src/strtou.c
src/unvis.c

Copyright:

Copyright 1980, 1982, 1986, 1989-1994

The Regents of the University of California. All rights reserved.

Copyright 1992 Keith Muller.

Copyright 2001 Mike Barcroft <mike@FreeBSD.org>

.

Some code is derived from software contributed to Berkeley by the American National Standards Committee X3, on Information Processing Systems.

.

Some code is derived from software contributed to Berkeley by Peter McIlroy.

.

Some code is derived from software contributed to Berkeley by Ronnie Kon at Mindcraft Inc., Kevin Lew and Elmer Yglesias.

.

Some code is derived from software contributed to Berkeley by Dave Borman at Cray Research, Inc.

.

Some code is derived from software contributed to Berkeley by Paul Vixie.

.

Some code is derived from software contributed to Berkeley by Chris Torek.

.

Copyright UNIX System Laboratories, Inc.

All or some portions of this file are derived from material licensed to the University of California by American Telephone and Telegraph Co. or Unix System Laboratories, Inc. and are reproduced herein with the permission of UNIX System Laboratories, Inc.

License: BSD-3-clause-Regents

Files:

src/vis.c

Copyright:

Copyright 1989, 1993

The Regents of the University of California. All rights reserved.

.

Copyright 1999, 2005 The NetBSD Foundation, Inc.

All rights reserved.

License: BSD-3-clause-Regents and BSD-2-clause-NetBSD

Files:

include/bsd/libutil.h

Copyright:

Copyright 1996 Peter Wemm <peter@FreeBSD.org>.

All rights reserved.

Copyright 2002 Networks Associates Technology, Inc.

All rights reserved.

License: BSD-3-clause-author

Files:

man/timeradd.3bsd

Copyright:

Copyright

2009 Jukka Ruohonen <jruohonen@iki.fi>

Copyright 1999 Kelly Yancey <kbyanc@posi.net>

All rights reserved.

License: BSD-3-clause-John-Birrell

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the author nor the names of any co-contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.

THIS SOFTWARE IS PROVIDED BY JOHN BIRRELL AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

man/setproctitle.3bsd

Copyright:

Copyright 1995 Peter Wemm <peter@FreeBSD.org>

All rights reserved.

License: BSD-5-clause-Peter-Wemm

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice immediately at the beginning of the file, without modification, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. This work was done expressly for inclusion into FreeBSD. Other use is permitted provided this notation is included.
4. Absolutely no warranty of function or purpose is made by the author Peter Wemm.
5. Modifications may be freely made to this file providing the above conditions are met.

Files:

include/bsd/stringlist.h

man/arc4random.3bsd

man/fmtcheck.3bsd

man/humanize_number.3bsd

man/stringlist.3bsd

man/timeval.3bsd

src/fmtcheck.c

src/humanize_number.c

src/stringlist.c

src/strtonum.c

Copyright:

Copyright 1994, 1997-2000, 2002, 2008, 2010, 2014

The NetBSD Foundation, Inc.

Copyright 2013 John-Mark Gurney <jmg@FreeBSD.org>

All rights reserved.

.

Copyright 2014

The NetBSD Foundation, Inc.

All rights reserved.

.

Some code was derived from software contributed to The NetBSD Foundation by Taylor R. Campbell.

.

Some code was contributed to The NetBSD Foundation by Allen Briggs.

.

Some code was contributed to The NetBSD Foundation by Luke Mewburn.

.

Some code is derived from software contributed to The NetBSD Foundation by Jason R. Thorpe of the Numerical Aerospace Simulation Facility, NASA Ames Research Center, by Luke Mewburn and by Tomas Svensson.

.

Some code is derived from software contributed to The NetBSD Foundation by Julio M. Merino Vidal, developed as part of Google's Summer of Code 2005 program.

.

Some code is derived from software contributed to The NetBSD Foundation by Christos Zoulas.

.

Some code is derived from software contributed to The NetBSD Foundation by Jukka Ruohonen.

License: BSD-2-clause-NetBSD

Files:

include/bsd/sys/endian.h
man/byteorder.3bsd
man/closefrom.3bsd
man/expand_number.3bsd
man/flopen.3bsd
man/getpeereid.3bsd
man/pidfile.3bsd
src/expand_number.c
src/pidfile.c
src/reallocf.c
src/timeconv.c

Copyright:

Copyright 1998, M. Warner Losh <imp@freebsd.org>

All rights reserved.

.

Copyright 2001 Dima Dorfman.

All rights reserved.

.

Copyright 2001 FreeBSD Inc.

All rights reserved.

.

Copyright 2002 Thomas Moestl <tmml@FreeBSD.org>

All rights reserved.

.

Copyright 2002 Mike Barcroft <mike@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Colin Percival

All rights reserved.

.

Copyright 2007 Eric Anderson <anderson@FreeBSD.org>

Copyright 2007 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2007 Dag-Erling Codan Smrgrav

All rights reserved.

.

Copyright 2009 Advanced Computing Technologies LLC

All rights reserved.

.

Copyright 2011 Guillem Jover <guillem@hadrons.org>

License: BSD-2-clause

Files:

src/flopen.c

Copyright:

Copyright 2007-2009

Dag-Erling Codan Smrgrav

All rights reserved.

License: BSD-2-clause-verbatim

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer in this position and unchanged.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/bsd/sys/tree.h
man/fparseln.3bsd
man/tree.3bsd
src/fparseln.c

Copyright:

Copyright 1997 Christos Zoulas.
All rights reserved.

.

Copyright 2002 Niels Provos <provos@citi.umich.edu>

All rights reserved.

License: BSD-2-clause-author

Files:

include/bsd/readpassphrase.h
man/readpassphrase.3bsd
man/strncpy.3bsd
man/strtonum.3bsd
src/arc4random.c
src/arc4random_linux.h
src/arc4random_uniform.c
src/arc4random_unix.h
src/arc4random_win.h
src/closefrom.c
src/freezero.c
src/getentropy_aix.c
src/getentropy_bsd.c
src/getentropy_hpux.c
src/getentropy_hurd.c
src/getentropy_linux.c
src/getentropy_osx.c
src/getentropy_solaris.c
src/getentropy_win.c
src/readpassphrase.c
src/reallocarray.c
src/reallocarray.c
src/strlcat.c
src/strncpy.c
test/explicit_bzero.c
test/strtonum.c

Copyright:

Copyright 2004 Ted Unangst and Todd Miller
All rights reserved.

.

Copyright 1996 David Mazieres <dm@uun.org>

Copyright 1998, 2000-2002, 2004-2005, 2007, 2010, 2012-2015

Todd C. Miller <Todd.Miller@courtesan.com>

Copyright 2004 Ted Unangst

Copyright 2004 Otto Moerbeek <otto@drijf.net>

Copyright 2008 Damien Miller <djm@openbsd.org>

Copyright 2008, 2010-2011, 2016-2017 Otto Moerbeek <otto@drijf.net>

Copyright 2013 Markus Friedl <markus@openbsd.org>

Copyright 2014 Bob Beck <beck@obtuse.com>

Copyright 2014 Brent Cook <bcook@openbsd.org>

Copyright 2014 Pawel Jakub Dawidek <pjd@FreeBSD.org>

Copyright 2014 Theo de Raadt <deraadt@openbsd.org>

Copyright 2014 Google Inc.

Copyright 2015 Michael Felt <aixtools@gmail.com>

Copyright 2015, 2022 Guillem Jover <guillem@hadrons.org>

License:

ISC

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/inet_net_pton.c

Copyright:

Copyright 1996 by Internet Software Consortium.

License: ISC-Original

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice

and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/setproctitle.c

Copyright:

Copyright 2010 William Ahern

Copyright 2012 Guillem Jover <guillem@hadrons.org>

License: Expat

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files:

src/explicit_bzero.c

src/chacha_private.h

Copyright:

None

License: public-domain

Public domain.

Files:

man/mdX.3bsd

Copyright:

None

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this

file. As long as you retain this notice you

can do whatever you want with this stuff. If we meet some day, and you think this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

License: BSD-3-clause-Regents

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause-author

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE

AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-author

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.47 aws-sdk-go-v2-service-sso 1.29.8

1.47.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer

```

    InterceptBeforeDeserialization
    AddTimeOffsetMiddleware
    RecordResponseTiming
    RequestResponseLogger
    InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //

```

This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

```

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {

```

```

return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err
= addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
}

```



```

if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint

```

InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"

```

```

"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde

```

```

}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```



```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)

```

```

if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger

```

OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
 "smithy": "2.0",
 "shapes": {


```

"com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
  "type": "structure",
  "members": {
    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
        }
      },
      "ActiveDirectoryType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
        "traits": {
          "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
        }
      },
      "IsSharedActiveDirectory": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
        }
      },
      "DomainNetworkSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
      }
    }
  },

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

"IdentityProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>The identity provider for the user.</p>",

    "smithy.api#required": {}
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
  }
},
"Tags": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {

```

```

        "target": "smithy.api#String"
    }
},
"traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
    "smithy.api#error": "server"
}

},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        },
        "Tags": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
            }
        }
    }
}

```

```

    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      }
    ]
  }
}

```



```

"target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ],
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  }
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#Directory": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ]
    }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {

```

```

        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},

"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
        "Subnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":

```

```

{
  "Attribute": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
    }
  },
  "Operation": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
    }
  },
  "Value": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Value of the filter.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
  },
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    }
  },

```



```

    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      }
    }
  }

```

```

    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
            "smithy.api#required": {}
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}$"
    }
  },

```

```

"com.amazonaws.licensemanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    }
  },

```

```

    "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager User Subscriptions",
    "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {

            "Region": {
                "builtIn": "AWS::Region",
                "required": false,
                "documentation": "The AWS region used to dispatch the request.",
                "type": "string"
            },
            "UseDualStack": {
                "builtIn": "AWS::UseDualStack",
                "required": true,
                "default": false,
                "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
                "type": "boolean"
            },
            "UseFIPS": {
                "builtIn": "AWS::UseFIPS",
                "required": true,
                "default": false,
                "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
                "type": "boolean"
            },
            "Endpoint": {
                "builtIn": "SDK::Endpoint",
                "required": false,
                "documentation": "Override the endpoint used to send this request",
                "type": "string"
            }
        },
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "isSet",
                        "argv": [
                            {
                                "ref": "Endpoint"
                            }
                        ]
                    }
                ]
            }
        ]
    }

```

```

    }

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],
        "endpoint": {
          "url": {
            "ref": "Endpoint"
          },
          },
        "properties": {},
        "headers": {}
      },
      {
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
}

```



```

    },
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Region"
            }
          ]
        }
      ]
    },
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "aws.partition",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          },
          ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
            ],
            true
          ]
        },
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            ],
            true
          ]
        }
      ],
    ],

```

```

        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            }
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    },
    {
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ]
            }
        ]
    }
]

```

```

    ],
    {
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {

```

```

        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    },
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

    ]
  }
  ]
}
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  },
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [],
  "endpoint": {
    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],

```

```

        "error": "Invalid Configuration: Missing Region",
        "type": "error"
    }
}
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-east-1",

"UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
}

```

```

        {
            "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.1.api.amazonwebservices.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
                }
            },
            "params": {
                "Region": "cn-north-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
                }
            }
        }
    ]
}

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
}

```



```

    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
      },
      "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
}

```

```

    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],

```

```

        "version": "1.0"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
        "IdentityProviderArn": {
            "target": "smithy.api#String",
            "traits": {

```

```

    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {

```

```

    "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",
          "value": "PROVISIONED",
          "documentation": ""
        }
      ]
    }
  }
}

```

```

        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",

```

```

"members": {
  "ServerType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>",
      "smithy.api#required": {}
    }
  },
  "ServerSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
    "traits": {
      "smithy.api#documentation": "for your server.<p>The <code>ServerSettings</code> resource contains the settings",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ]
}

```



```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of InstanceSummary resources that contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {

        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"

      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },

    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>"
            }
        }
    }
}

```

```

    </p>"
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags

```

```

you want to retrieve.</p>",
    "smithy.api#httpLabel": {},
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        }
    }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {

```

```

    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    }
  }

```

```

    }
  },
  "SubscriptionStartDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The start date of a subscription.</p>"
    }
  },
  "SubscriptionEndDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The end date of a subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
  "type": "structure",
  "members": {
    "RdsSalCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
  {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
  },
  "Settings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",

```



```

    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
  "type": "union",

  "members": {
    "RdsSalSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RDS_SAL",
        "value": "RDS_SAL",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException": {
  "type": "structure",
  "members": {
    "message": {

```

```

        "target": "smithy.api#String"
    }
},
"traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
        "Subnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        }
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
    },
  },
  "uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
  },
},

```

```

"IdentityProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"Product": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
    "smithy.api#required": {}
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
  }
},
"Tags": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        }
    },
    "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    },
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  },
},
"traits": {
  "smithy.api#input": {}
},
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
},
"traits": {
  "smithy.api#output": {}
},
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    }
  },
},

```

```

    "RemoveSubnets": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
        "traits": {
            "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
            "smithy.api#required": {}
        }
    },
    "SecurityGroupId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    }
},

    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err

```

```

error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
    if err = addClientUserAgent(stack, options); err != nil {

```



```

return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}

```

```

}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type

```

```

GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }

```

```

}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}

```

```

}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:
        region,
    ServiceID:  ServiceID,
    OperationName: "GetLicense",
}
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                path: sdkbase
                ref: ${ github.base_ref }
            - name: Checkout this ref
              uses: actions/checkout@v2
              with:
                path: new-ref
                fetch-depth: 0
            - name: Get Diff
              run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
                ${ github.sha } > refDiffFiles.txt
            - name: Get Target Files
              run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
            - name: Checkout scancode
              uses: actions/checkout@v2
              with:
                repository: nexB/scancode-toolkit

```

```

    path: scancode-toolkit
    fetch-depth: 1
  - name: Set up Python
    ${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```



```

}

out := result.(*CheckoutLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.

```

CheckoutType types.CheckoutType

// Allowed license entitlements.

EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.

LicenseArn

*string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

return err

}

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)

if err != nil {

return err

}

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)

if err != nil {

return err

}

if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {

return fmt.Errorf("add protocol finalizers: %v", err)

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

return err

```

}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license for offline use.

```

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```

```

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package grafana
```

```
import (
    "context"
```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].

```

```

//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {

```

```

return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptExecution(stack,
options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
return err
}
if err = addInterceptTransmit(stack, options); err != nil {
return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
}

```

```

if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*DeleteLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

```

```

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

```

```

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```



```

if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}

```

```

if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckInLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {
```

```
    if params == nil {  
        params = &CheckInLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
    c.addOperationCheckInLicenseMiddlewares)
```

```
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata
```

```

    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }

```

```

}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {

```



```

    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware

```

RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
 "type": "integer"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
 "type": "long"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
 "type": "operation",
 "input": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
 },
 "output": {
 "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
 },
 "errors": [

```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/DeregisterSubscriptionProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
            }
        },
        "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

```



```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/GetServiceSettings"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  }
}

```

```

    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",

```



```

    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },

```

```

    "SubscriptionProviderCreateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
      }
    },
    "SubscriptionProviderUpdateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
      }
    },
    "DualSubscription": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
      }
    },
    "RegisteredWithSubscriptionProvider": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      }
    },

```

```

    "traits": {
      "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
      }
    ],
    "traits": {
      "aws.api#service": {
        "sdkId": "License Manager Linux Subscriptions",

```

```

    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
      },
      "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",

```

```

        "type": "Boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        },
        true
    ]
}

    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [

```

```

        {
            "ref": "Region"
        }
    ],
    "assign": "PartitionResult"
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            },
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            }
                        ]
                    }
                ]
            },
            {
                "fn": "supportsFIPS"
            }
        ]
    }
]

```

```

    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],

"type": "tree"
},
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
}
],

```



```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsFIPS"
                                ]
                            },
                            true
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "rules": [
                            {
                                "conditions": [],
                                "endpoint": {
                                    "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                                "properties": {},
                                "headers": {}

```

```

        },
        "type": "endpoint"
    }
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
},
],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}

```



```

        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        },
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
                }
            }
        },
        {
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": false
            }
        }
    ],
    {

```

```

    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },

```

```

    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {

```

```

        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
{
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",

```



```

    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,

```

```

        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {

```

```

    "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Instances"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {

```

```

    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                }
            }
        },
        "traits": {
            "smithy.api#documentation":
                "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n\t\torganization, the returned results will include data aggregated across your accounts in\n\t\tOrganizations.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptions"
      }
    }
  }
}

```

```

    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
    "type": "structure",
    "members": {
      "Filters": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

        "traits": {
          "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n                <code>contains</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>>equals</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>Notequal</code>\n            </p>\n        </li>\n    </ul>"
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 16384
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "NextToken

```

length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."

```
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
```

```

        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderSources": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
    "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
        }
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>"
        }
    }
},

"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse": {

```



```

    "smithy.api#readonly": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to
list metadata tags.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        }
      ]
    }
  }
}

```

```

    },
    {
      "name": "CONTAINS",
      "value": "Contains",
      "documentation": "Contains operator"
    }
  ],
  "smithy.api#length": {
    "min": 1,
    "max": 20
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    }
  ]
}

```



```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
        }
      },
      "SubscriptionProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux subscription provider \n\t\t\t\tthat you requested.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
        }
      }
    }
  }
}

```



```

"members": {
  "message": {
    "target": "smithy.api#String"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
  "smithy.api#error": "client"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "RedHat",
        "value": "RedHat",
        "documentation": "RedHat subscription provider namespace"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      }
    ],

```

```

        {
            "name": "INVALID",
            "value": "INVALID",
            "documentation": "INVALID status"
        },
        {
            "name": "PENDING",
            "value": "PENDING",
            "documentation": "PENDING status"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {

```



```

        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",

  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tagKeys": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {

"target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
{}
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
                "smithy.api#required": {}
            }
        },
        "AllowUpdate": {
            "target": "smithy.api#Boolean",
            "traits": {
                "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
            }
        }
    }
}
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },

```

```

    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.48 google-uuid 1.6.0

1.48.1 Available under license :

Paul Borman <borman@google.com>

bmatsuo

shawnps

theory

jboverfelt

dsymonds

cd1

wallclockbuilder

dansouza

Copyright (c) 2009,2014 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.49 aws-sdk-go-v2-aws-protocol-eventstream 1.7.2

1.49.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

```

InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```



```

)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //

```

```

// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

```

```

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}

```

```

if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}

```

```

}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

```

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader


```

ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

```

```

type ListReceivedLicensesInput struct {

```

```

    // Filters to scope the results. The following filters are supported:

```

```

    //

```

```

    // - ProductSKU

```

```

    //

```

```

    // - Status

```

```

    //

```

```

    //

```

```

    // - Fingerprint

```

```

    //

```

```

    // - IssuerName

```

```

    //

```

```

    // - Beneficiary

```

```

    Filters []types.Filter

```

```

    // Amazon Resource Names (ARNs) of the licenses.

```

```

    LicenseArns []string

```

```

    // Maximum number of results to return in a single call.

```

```

    MaxResults *int32

```

```

    // Token for the next set of results.

```

```

    NextToken *string

```

```

    noSmithyDocumentSerde

```

```

}

```

```

type ListReceivedLicensesOutput struct {

```

```

    // Received license details.

```

```

    Licenses []types.GrantedLicense

```

```

    // Token for the next set of results.

```

```

    NextToken *string

```

```

    // Metadata pertaining to the operation's result.

```

```

    ResultMetadata middleware.Metadata

```

```

    noSmithyDocumentSerde

```

```

}

```

```

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err

```

```

error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
    if err =

```

```

addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters

```

```

[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
}

```



```

if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

DisassociateLicense
Initialize stack step
    spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
    ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware

```

```

RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {

```

```

    "DirectoryId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
      "traits": {
        "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
      }
    },
    "ActiveDirectorySettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    },
    "IsSharedActiveDirectory": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4

```

addresses that are used for the Active Directory.</p>",

```

    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  },
  "DomainIpv6List": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Ipv6List",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 2
      }
    }
  },
  "DomainCredentialsProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
    }
]
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {

```

"smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as Pending billing status) in Amazon Web Services Billing. For more information, see Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based subscription.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains information for the user to associate.</p>"
        }
      }
    }
  },
```



```

    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
      "type": "structure",
      "members": {
        "InstanceUserSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active

```

Directory that contains user identity details.</p>",

```
    "smithy.api#required": {}
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
```

```

"members": {
  "SecretsManagerCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>  
resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        }
      ],
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/DeregisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
    "type": "structure",
    "members": {

```

```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Directory": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
      }
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {

        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",

```



```

"members": {
  "Username": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",

```

```

        "smithy.api#required": {}
    }
}
},

"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
        "Subnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
                "smithy.api#length": {
                    "min": 1
                },
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members":
{
    "Attribute": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
    },
    "Operation": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
    },
    "Value": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
    }
}
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  }
},
"Settings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

```

```

        "smithy.api#required": {}
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
        }
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of the identity provider.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
    },
    "FailureMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
        }
    },

```

```

"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The IdentityProvider resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The IdentityProvider resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
        }
    }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4})::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}):{1,7}|([0-9a-fA-F]{1,4}):{1,6}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IPv6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IPv6"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
        }
      ]
    }
  }

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],

```

```

"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
  "UseDualStack": {
    "builtIn": "AWS::UseDualStack",

```

```

        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
]
],

```

```

        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [

{
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
    },
    "rules": [
        {
            "conditions": [

```

```

    {
      "fn": "aws.partition",
      "argv": [
        {
          "ref": "Region"
        }
      ],
      "assign": "PartitionResult"
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            }
          ],
          true
        }
      ],
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseDualStack"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  }
                ],
                "supportsFIPS"
              }
            ]
          }
        ]
      }
    ]
  ]
}

```

```

        ]
      }
    ]
  },

  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
]
}
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    {
      "ref": "UseFIPS"
    },
    true
  ]
},
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          },
          true
        ]
      }
    ],
    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
    "properties": {},
    "headers": {}
  },
  {
    "type": "endpoint"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled but this partition does not support FIPS",

```



```

        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },

```

```

    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",

"UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
      "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
    },

```

```

        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },

```

```

        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
}

```

```

    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    }
  },

```

```

    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license
server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The

```



```

<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {

```

```

    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
    "type": "string"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",
          "value": "PROVISIONED",
          "documentation": ""
        },
        {
          "name": "DELETING",
          "value": "DELETING",
          "documentation": ""
        },
        {
          "name": "DELETION_FAILED",
          "value": "DELETION_FAILED",
          "documentation": ""
        },
        {
          "name": "DELETED",
          "value": "DELETED",
          "documentation": ""
        }
      ]
    }
  }
}

```

```

    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      },
      {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        }
    },

```

```

    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,

```

```

        "method": "POST",
        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",
    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that

```

contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
```



```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoints": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single

```

```

request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
        "smithy.api#http": {
            "method": "GET",
            "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#readonly": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
        "type": "structure",
        "members": {
            "ResourceArn": {
                "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                    "smithy.api#httpLabel": {},
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        }
    }
}

```

```

    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    }
  },

```



```

    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a product for this user.</p>",
        "smithy.api#required": {}
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    }
  },

```

```

    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {

```

```

    "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {

```

```

    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
        "type": "union",

        "members": {
            "RdsSalSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
    {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {

```

```

        "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
        "smithy.api#length": {
            "min": 1
        },
        "smithy.api#required": {}
    }

},
    "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
            "smithy.api#required": {}
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
        "type": "operation",
        "input": {
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
            },
            "output": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
                },
                {
                    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
                },
                {
                    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
                },
                {
                    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
                },
                {
                    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
                }
            ]
        }
    }
}

```

```

    },

    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
                "smithy.api#required": {}
            }
        }
    }
}

```



```

    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {

```

```

        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "Tags": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
            "traits": {
                "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {}
    }
}

```

```

        "min": 0,
        "max": 50
    },
    "smithy.api#sensitive": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",

```

```

"members": {
  "ResourceArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
      "smithy.api#httpLabel": {},
      "smithy.api#required": {}
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p></li> <li><p>Subnets which you want to remove the VPC endpoints from.</p></li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p></li> </ul>",
      "smithy.api#required": {}
    }
  }
}

```



```

    }

    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    }
}

```

```

    },

    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Removes the Grafana Enterprise license from a workspace.
```

```

func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```
    out := result.(*DisassociateLicenseOutput)
```

```

out.ResultMetadata = metadata
return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");

```

```

err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {

```

```

return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptExecution(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}

```

```

}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

```

)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }

```

```

}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{ }, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {

```



```

    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}

```

```

    }
    if err = addInterceptBeforeSerialization(stack,
options); err != nil {
        return err
    }
    if err = addInterceptAfterSerialization(stack, options); err != nil {
        return err
    }
    if err = addInterceptBeforeSigning(stack, options); err != nil {
        return err
    }
    if err = addInterceptAfterSigning(stack, options); err != nil {
        return err
    }
    if err = addInterceptTransmit(stack, options); err != nil {
        return err
    }
    if err = addInterceptBeforeDeserialization(stack, options); err != nil {
        return err
    }
    if err = addInterceptAfterDeserialization(stack, options); err != nil {
        return err
    }
    if err = addSpanInitializeStart(stack); err != nil {
        return err
    }
    if err = addSpanInitializeEnd(stack); err != nil {
        return err
    }
    if err = addSpanBuildRequestStart(stack); err != nil {
        return err
    }
    if err = addSpanBuildRequestEnd(stack); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

```

on: [pull_request]

permissions:
contents: read

jobs:

licensescan:

name: License Scan
runs-on: ubuntu-latest
strategy:
matrix:
python-version: [3.9]

steps:

- name: Checkout target
uses: actions/checkout@v2
with:
path: sdkbase
ref: \${{ github.base_ref }}
- name: Checkout this ref
uses: actions/checkout@v2
with:
path: new-ref
fetch-depth: 0
- name: Get Diff
run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT \${{ github.event.pull_request.base.sha }}
\${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
uses: actions/checkout@v2
with:
repository: nexB/scancode-toolkit
path: scancode-toolkit
fetch-depth: 1
- name: Set up Python
\${{ matrix.python-version }}
uses: actions/setup-python@v2
with:
python-version: \${{ matrix.python-version }}

ScanCode

- name: Self-configure scancode
working-directory: ./scancode-toolkit
run: ./scancode --help
- name: Run Scan code on target
run: cat targetFiles.txt | while read filename; do echo ./sdkbase/\$filename; done | xargs ./scancode-toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt

```

- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
  are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
    c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type CheckoutLicenseInput struct {

    //
    CheckoutType types.CheckoutType
    // This member is required.
    CheckoutType types.CheckoutType
}

```

```

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn

```

```

*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```

```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}

```



```

}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)

```

```

out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutBorrowLicenseOutput struct {

```

```

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

    // Allowed license entitlements.
    EntitlementsAllowed

```

[]types.EntitlementData

// Date and time at which the license checkout expires.

Expiration *string

// Date and time at which the license checkout is issued.

IssuedAt *string

// Amazon Resource Name (ARN) of the license.

LicenseArn *string

// License consumption token.

LicenseConsumptionToken *string

// Node ID.

NodeId *string

// Signed token.

SignedToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
```

```
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
```

```
    }
```

```
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return
```

```
    err
```

```
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
```

```
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
```

```
    }
```

```
    if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```

return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
}

```

```

if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
    err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-

```

Grafana-Enterprise.html

```
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }
```

```
    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
```

```
    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}
```

type AssociateLicenseInput struct {

```
    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType
```

```
    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string
```

```
    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
    Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string
```

```
    noSmithyDocumentSerde
}
```

type AssociateLicenseOutput
struct {

```
    // A structure containing data about the workspace.
    //
```

```

// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
        options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {

```



```

    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}

```

```

}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

```

```

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}

```

```

}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),

```

```

middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}

```

```

}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckInLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{}    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
c.addOperationCheckInLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }  
}
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckInLicenseInput struct {
```

```
    // License consumption token.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseConsumptionToken *string
```

```
    // License beneficiary.
```

```
    Beneficiary *string
```

```

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}

```

```

}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}

```

```

}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID:
        ServiceID,
    OperationName: "CheckInLicense",
}
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

```

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ]
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  }
}

```



```

    }
  },
  "Operator": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
    "traits": {
      "smithy.api#documentation": "<p>An operator for filtering results.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
      "smithy.api#http": {

```



```

        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation":
                "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
        "type": "structure",
        "members": {
            "SubscriptionProviderArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
                }
            },
            "SubscriptionProviderSource": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
                "traits": {
                    "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
                }
            },
            "SecretArn": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
                }
            }
        },
    },

```



```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated

```

```

data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

```

```

    "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n    <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n        operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
},
    "ProductCode": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
        "traits": {
            "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n        operation values</a> in the <i>License Manager User Guide</i> .</p>"
        }
    },
    "LastUpdatedTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
        }
    },
    "SubscriptionName": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
        }
    },
    "OsVersion": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
        }
    },
    "SubscriptionProviderCreateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
        }
    },
    "SubscriptionProviderUpdateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
        }
    }
}

```



```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",

```

```

        "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running\n    Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",

```



```

        "argv": [
            {
                "ref": "Endpoint"
            }
        ]
    },
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                        true
                    ]
                }
            ]
        }
    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    }
}

```

```

        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [

"fn": "aws.partition",
            "argv": [
                {
                    "ref": "Region"
                }
            ],
            "assign": "PartitionResult"
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {

```

```

    "ref": "UseFIPS"
    },
    true
  ]
},
{
  "fn": "booleanEquals",
  "argv": [
    {
      "ref": "UseDualStack"
    },
    true
  ]
}

],

"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              }
            ]
          }
        ]
      },
      {
        "fn": "supportsFIPS"
      }
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
]

```

```

        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],

"rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
  "type": "tree"
},
  {
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
    "type": "error"
  },
  "type": "tree"
},
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ]
  }

```

```

    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "rules": [
              {
                "conditions": [],
                "endpoint": {
                  "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                  "properties": {},
                  "headers": {}
                },
                "type": "endpoint"
              }
            ],
            "type": "tree"
          }
        ],
        "type": "tree"
      },
      {
        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
      }
    ]
  }
}

```

```

    }
  ],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},

```

```

        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"

```

```

    }
  ]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",

```



```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {

```

```

        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

```

```

{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips enabled and dualstack disabled",
  "expect": {
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
  },
  "params": {

"Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
  "expect": {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "Missing region",

  "expect": {
    "error": "Invalid Configuration: Missing Region"
  }
}
],
"version": "1.0"

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",

```

```

"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n  subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptionInstances"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Instances"
  }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n  \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n  \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n  \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n  \tto
streamline results:</p>\n
<ul>\n
  <li>\n
    <p>\n
      <code>AccountID</code>\n
    </p>\n
    </li>\n
    <li>\n
      <p>\n
        <code>AmiID</code>\n
      </p>\n
      </li>\n
    <li>\n
      <p>\n
        <code>DualSubscription</code>\n
      </p>\n
      </li>\n
  </ul>

```

```

</li>\n      <p>\n          <code>InstanceID</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>InstanceType</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>ProductCode</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>Region</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>Status</code>\n      </p>\n  </li>\n  <li>\n      <p>\n          <code>UsageOperation</code>\n      </p>\n  </li>\n</ul>\n<p>For each filter, you can use one of the following <code>Operator</code> values to \n \t\define the behavior of the filter:</p>\n  <ul>\n      <li>\n          <p>\n              <code>contains</code>\n          </p>\n      </li>\n      <li>\n          <p>\n              <code>equals</code>\n          </p>\n      </li>\n      <li>\n          <p>\n              <code>Notequal</code>\n          </p>\n      </li>\n  </ul>\n  }\n},\n\"MaxResults\": {\n    \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger\",\n    \"traits\": {\n        \"smithy.api#documentation\": \"<p>The maximum items to return in a request.</p>\"\n    }\n},\n\"NextToken\": {\n    \"target\": \"smithy.api#String\",\n    \"traits\": {\n        \"smithy.api#documentation\": \"<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>\",\n        \"smithy.api#length\": {\n            \"min\": 1,\n            \"max\": 16384\n        }\n    }\n},\n\"traits\": {\n    \"smithy.api#documentation\":\n\"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases.\"\n},\n},\n\"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse\": {\n    \"type\": \"structure\",\n    \"members\": {\n        \"Instances\": {\n            \"target\": \"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList\",\n            \"traits\": {\n                \"smithy.api#documentation\": \"<p>An array that contains instance objects.</p>\"\n            }\n        }\n    },\n    \"NextToken\": {\n        \"target\": \"smithy.api#String\",\n        \"traits\": {\n            \"smithy.api#documentation\": \"<p>The next token used for paginated responses. When this \n\tfield

```


isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional

objects.

```
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
```

<p>Lists the Linux subscriptions that have been discovered. If you have linked your organization, the returned results will include data aggregated across your accounts in Organizations.</p>

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

    "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n                <code>contains</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>equals</code>\n            </p>\n        </li>\n        <li>\n            <p>\n                <code>NotEqual</code>\n            </p>\n        </li>\n    </ul>"
    }
},
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
},
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",

```

```
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
    }
  }
},
```

"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {

```
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
```

```
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
```

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
```

```
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
```



```

    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "tags": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
      "smithy.api#http": {
        "code": 200,

```



```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,

```

```

\n\t\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-

```

```

\\.{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
      "type": "structure",
      "members": {
        "Name": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the subscription.</p>"
          }
        },
        "Type": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
          }
        },
        "InstanceCount": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
          "traits": {
            "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        },
        {
          "name": "PENDING",
          "value": "PENDING",
          "documentation": "PENDING status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":

```

```

{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```



```

        "target": "smithy.api#String"
    }
},
"traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
            "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input":
    {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },

```

```

    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":
{}
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated

```

```

from.</p>"
    }

    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.50 go-urn 1.4.0

1.50.1 Available under license :

MIT License

Copyright (c) 2018 Leonardo Di Donato

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.51 go.yaml.in-yaml 2.4.3

1.51.1 Available under license :

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.52 x-net 0.46.0

1.52.1 Available under license :

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.53 aws-sdk-go-v2-credentials 1.18.19

1.53.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd


```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }
}

```

```

result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.

```

```

//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

```

```
}  
}  
CheckoutBorrowLicense  
Initialize stack step  
  spanInitializeStart  
  InterceptExecution  
  RegisterServiceMetadata  
  legacyEndpointContextSetter  
  SetLogger  
  OperationInputValidation  
  spanInitializeEnd  
Serialize stack step  
  spanBuildRequestStart  
  setOperationInput  
  ResolveEndpoint  
  InterceptBeforeSerialization  
  OperationSerializer  
  InterceptAfterSerialization  
Build stack step  
  ClientRequestID  
  ComputeContentLength  
  SetCredentialSourceMiddleware  
  UserAgent  
  AddTimeOffsetMiddleware  
  RecursionDetection  
  spanBuildRequestEnd  
Finalize stack step  
  spanRetryLoop  
  InterceptBeforeRetryLoop  
  Retry  
  InterceptAttempt  
  RetryMetricsHeader  
  ResolveAuthScheme  
  GetIdentity  
  ResolveEndpointV2  
  disableHTTPS  
  ComputePayloadHash  
  setLegacyContextSigningOptions  
  InterceptBeforeSigning  
  Signing  
  InterceptAfterSigning  
Deserialize stack step  
  AddRawResponseToMetadata  
  ErrorCloseResponseBody  
  CloseResponseBody  
  ResponseErrorWrapper  
  RequestIDRetriever  
  InterceptAfterDeserialization
```

OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody

ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning

```

Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //

```

```

// - ProductSKU
//
// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},

```

```

middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

```

DeleteLicense

```

```

Initialize stack step

```

```

spanInitializeStart

```

```

InterceptExecution

```

```

RegisterServiceMetadata

```

```

legacyEndpointContextSetter

```

```

SetLogger

```

```

OperationInputValidation

```

```

spanInitializeEnd

```

```

Serialize stack step

```

```
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licensemanager
```

```
import (
    "context"
```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

```



```

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}

```

```

}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

DisassociateLicense
Initialize stack step
    spanInitializeStart
    InterceptExecution
    RegisterServiceMetadata
    legacyEndpointContextSetter
    SetLogger
    OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
    setOperationInput
    ResolveEndpoint
    InterceptBeforeSerialization
    OperationSerializer
    InterceptAfterSerialization
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
    RecursionDetection
    spanBuildRequestEnd
Finalize stack step
    spanRetryLoop
    InterceptBeforeRetryLoop
    Retry
    InterceptAttempt
    RetryMetricsHeader
    ResolveAuthScheme
    GetIdentity
    ResolveEndpointV2

```

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    }
  },
  "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains

```

details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"

```
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
```

"smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"

```
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
```

```
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
    },
```

```
    "DomainIpv4List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
```

```
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
```

```
    "DomainIpv6List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
```

```

Directory.</p>",
    "smithy.api#length": {
        "min": 1,
        "max": 2
    }
},
"DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
},
"DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "SELF_MANAGED",
                "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-

```

```

zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      },
    }
  }
}

```



```

        "smithy.api#idempotent": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
    {
        "type": "structure",
        "members": {
            "Username": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                    "smithy.api#required": {}
                }
            },
            "InstanceId": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                    "smithy.api#required": {}
                }
            },
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The identity provider for the user.</p>",
                    "smithy.api#required": {}
                }
            },
            "Domain": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
                }
            },
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

```

```

"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {

```

```

    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid"
      }
    }
  }
}

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^([a-z0-9-]{10})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },

```



```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",

```

```

    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
        "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    },
    "Products": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server",
        "smithy.api#httpError": 500
      }
    }

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*"
      ]
    }
  }
}

```

```

    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"
  },
  "UseDualStack": {
    "builtIn": "AWS::UseDualStack",
    "required": true,
    "default": false,
    "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
    "type": "boolean"
  },
  "UseFIPS": {
    "builtIn": "AWS::UseFIPS",
    "required": true,
    "default": false,
    "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error."

```

```

        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    }
]

```

```

        ]
    }
},
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
},
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
            ],
        },
    ],
    true
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
    ],
    true
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                        ],
                        "supportsFIPS"
                    }
                ],
            }
        ],
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
}

]
}
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            },
            {
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    }
                ]
            }
        ]
    }

```



```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
},
],
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"

```

```

        },
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
    },

```

```

        "params": {
            "Region": "us-east-1",

"UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,

            "UseDualStack": true
        }
    }

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS":
false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },

```

```

{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint":
{
      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```

```

    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    }
  }
}

```



```

    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>"
      }
    },
    "ServerEndpoint": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  }
}

```

```

    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "PROVISIONING",
          "value": "PROVISIONING",
          "documentation": ""
        },
        {
          "name": "PROVISIONING_FAILED",
          "value": "PROVISIONING_FAILED",
          "documentation": ""
        },
        {
          "name": "PROVISIONED",
          "value": "PROVISIONED",
          "documentation": ""
        },
        {
          "name": "DELETING",
          "value": "DELETING",
          "documentation": ""
        },
        {
          "name": "DELETION_FAILED",
          "value": "DELETION_FAILED",
          "documentation": ""
        },
        {
          "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "HEALTHY",
          "value": "HEALTHY",
          "documentation": ""
        },
        {

```

```

        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
    },
    {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single

```

```

request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstances": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/instance/ListInstances"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceSummaries"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {

```

```

    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
      "InstanceSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoints": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits":

```

```

{
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
        }
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
    ]

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",

```



```

    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",

```

```

        "smithy.api#required": {}
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {

```

```

"type": "structure",
"members": {
  "RdsSalCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>The CredentialsProvider resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
      "smithy.api#required": {}
    }
  },
},
"traits": {
  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
},

```

```

"traits": {
  "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/RegisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",

```

```

        "smithy.api#length": {
            "min": 1
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services

```


Billing. For more information, see <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html> Viewing your monthly charges in the *Amazon Web Services Billing User Guide*.

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",

"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  }
}
```

```

        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        },
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
        ],
    },

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
      }
    },
    "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want

```

```

to tag.</p>",
    "smithy.api#httpLabel": {},
    "smithy.api#required": {}
  }
},
"Tags": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"
        }
    }
}

```



```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
      }
    },
    "UpdateSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
      "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",

```

```

        "traits": {
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    }
}

```

```

    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //

```

```

The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {

```

```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

}

out := result.(*GetLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```



```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);

```

```

err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:

```

```

python-version: [3.9]

steps:
  - name: Checkout target
    uses: actions/checkout@v2
    with:
      path: sdkbase
      ref: ${ github.base_ref }
  - name: Checkout this ref
    uses: actions/checkout@v2
    with:
      path: new-ref
      fetch-depth: 0
  - name: Get Diff
    run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
  - name: Get Target Files
    run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
  - name: Checkout scancode
    uses: actions/checkout@v2
    with:
      repository: nexB/scancode-toolkit
      path: scancode-toolkit
      fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

```

```

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

```

```
noSmithyDocumentSerde
}
```

```
func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
}
```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{

```

```

    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
  }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Checks out the specified license for offline use.
```

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

```

```

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

```

```

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}

```

```

}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```



```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "AssociateLicense",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

```

```

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```

```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {

```

```

return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
return err
}
if err = addInterceptExecution(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
return err
}
if err = addInterceptTransmit(stack, options); err != nil {
return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

```

CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

```


Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
}

```

```

if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}

```

CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {

```

```

    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
}

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```

```

    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",

```



```

"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    }

],

    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

],

    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],
"type": "tree"

}

],

```

```

    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
]

```

```

        },
        true
    ]
}

],

"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        }
                    ]
                }
            ],
            "supportsFIPS"
        }
    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
},
{
    "conditions": [],
}
],

"rules": [
{

```

```

        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

```



```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "error": "DualStack is enabled but this partition does not support DualStack",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
}
],
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"

```



```

    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  }
}

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
}

```

```

    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },

```

```

        {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Instances"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
        <ul>\n
          <li>\n
            <p>\n
              <code>AccountID</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>AmiID</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>DualSubscription</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>InstanceID</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>InstanceType</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>ProductCode</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>Region</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>Status</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>UsageOperation</code>\n
            </p>\n
          </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \t\define the behavior of the filter:</p>\n
        <ul>\n
          <li>\n
            <p>\n
              <code>contains</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>equals</code>\n
            </p>\n
          </li>\n
          <li>\n
            <p>\n
              <code>Notequal</code>\n
            </p>\n
          </li>\n
        </ul>"
      }
    }
  },

```



```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
        "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
    returned results will include data aggregated across your accounts in\n  Organizations.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptions"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Subscriptions"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
        that match one or\n  more sets of key-value pairs that you specify. For example, you can filter by the name of\n
        <code>Subscription</code> with an optional operator to see subscriptions that match,\n  partially match, or don't
        match a certain subscription's name.</p>\n  <p>The valid names for this filter are:</p>\n  <ul>\n
        <li>\n  <p>\n  <code>Subscription</code>\n  </p>\n  </li>\n  </ul>\n
        <p>The valid Operators for this filter are:</p>\n  <ul>\n  <li>\n  <p>\n  <code>contains</code>\n  </p>\n  </li>\n  <li>\n  <p>\n  <code>equals</code>\n  </p>\n  </li>\n  <li>\n  <p>\n
      
```

```

<code>Notequal</code>\n          </p>\n          </li>\n        </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",

```

```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSources": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
        }
      },
      "MaxResults": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {

```

```

"name": "Disabled",
  "value": "Disabled",
  "documentation": "Disabled OrganizationIntegration"
}
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {

```



```

subscription provider \n\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      }
    },
  },

```

```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",

```

```

        "documentation": "InProgress status"
    },
    {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [

```

```

        {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
    },

```


"smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are formatted as key value pairs.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
```



```

"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
  "smithy.api#http": {
    "method": "DELETE",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",

  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tagKeys": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input":
    {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",

```

```

        "smithy.api#required":
    {}
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
        }
    }
},
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
                }
            },
            "LinuxSubscriptionsDiscoverySettings": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
                "traits": {
                    "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
                }
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        }
    }
}

```

```

    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensors and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

InterceptAfterDeserialization

OperationDeserializer

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.54 aws-sdk-go-v2-service-dynamodb 1.52.2

1.54.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash

```

setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.
    LicenseName *string

    // Product name.
    //
    // This member is required.
    ProductName *string

    // Product SKU.
    //

```

```

// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},

```

```

middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //

```

```

// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err =
addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength

```

```

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))

```

```

(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.

```

NextToken *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
```

```
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack,
```

```
        options, "ListLicenses"); err != nil {
```

```
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addSetLoggerMiddleware(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addClientRequestID(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addComputeContentLength(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addComputePayloadSHA256(stack); err != nil {
```

```
        return err
    }
```

```
    if err = addRetry(stack, options); err != nil {
```

```
        return err
    }
```

```
    if err = addRawResponseToMetadata(stack); err != nil {
```

```
        return err
    }
```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {

```

```
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
```

DisassociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    }
  }
}

```



```

    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {

```

```

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
        "smithy.api#required": {}
      }
    },
    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```



```

    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

"target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
        }
    }
}

```

```

    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
}

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
{
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
        "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Settings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }

  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Describes an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Products": {

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that

```

contains the user information for the product subscription.</p>"

```
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV4List": {
```

```

    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:[1,7]:|([0-9a-fA-F]{1,4}:[1,6]:[0-9a-fA-F]{1,4})$)"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",

```

```

        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],

```

```

        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                ],
            },
        ],
    },

```


true

```
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          }
        ]
      },
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  }
]
```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",

```



```

    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsDualStack"
              ]
            }
          ]
        }
      ]
    },
    {
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        },
        {
          "type": "tree"
        },
        {
          "conditions": [],
          "error": "DualStack is enabled but this partition does not support DualStack",
          "type": "error"
        }
      ],
      "type": "tree"
    },
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            }
        }
    ]
}

```

```

    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
}

```

```

    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint":
{
          "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"

```



```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
  },

```

```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
      },
      "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>"
        }
      }
    }
  }
}

```

```

resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {

```

```

        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
    },
    {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
    },
    {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            }
        ]
    }
}

```

```

        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```



```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results

```

to return from a single request.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        }
    }

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
      }
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```



```

    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        },
        "traits": {
          "smithy.api#output": {}
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output":
  {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for this user.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a

```

reference to the credentials provider that's used for RDS license server user administration.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {

  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},

```



```

    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
      "Endpoint": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
    "type": "string",

```

```

    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings

```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

"uri": "/user/StartProductSubscription"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

```

```

    "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {

```



```

        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",

```

```

"members": {
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
  }
},
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

```

    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

```

```

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}

```



```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if

```

```

err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err

```

```

!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                  path: sdkbase
                  ref: ${{ github.base_ref }}

```

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.

```



```

ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

type CheckoutBorrowLicenseInput struct {

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//

the request.

//

// This member is required.

ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)

// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].

//

// [RFC 7518 Digital Signature with RSASSA-PSS]: <https://tools.ietf.org/html/rfc7518#section-3.5>

//

// This member is required.

DigitalSignatureMethod types.DigitalSignatureMethod

```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.

```

```

//
// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
}

```

```

if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,

```

```

c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

)

```
// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}
```

```
type DeleteLicenseInput struct {
```

```
    // Amazon Resource Name (ARN) of the license.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseArn *string
```

```
    // Current version of the license.
```

```
    //
```

```
    // This member is required.
```

```
    SourceVersion
```

```
    *string
```

```
    noSmithyDocumentSerde
```

```
}
```

```
type DeleteLicenseOutput struct {
```

```
    // Date when the license is deleted.
```

```
    DeletionDate *string
```

```
    // License status.
```

```
    Status types.LicenseDeletionStatus
```

```
    // Metadata pertaining to the operation's result.
```

```
    ResultMetadata middleware.Metadata
```

```
    noSmithyDocumentSerde
```

```

}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```



```

}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
}
}

```

```
    OperationName: "DeleteLicense",  
  }  
}  
CheckInLicense  
Initialize stack step  
  spanInitializeStart  
  InterceptExecution  
  RegisterServiceMetadata  
  legacyEndpointContextSetter  
  SetLogger  
  OperationInputValidation  
  spanInitializeEnd  
Serialize stack step  
  spanBuildRequestStart  
  setOperationInput  
  ResolveEndpoint  
  InterceptBeforeSerialization  
  OperationSerializer  
  InterceptAfterSerialization  
Build stack step  
  ClientRequestID  
  ComputeContentLength  
  SetCredentialSourceMiddleware  
  UserAgent  
  AddTimeOffsetMiddleware  
  RecursionDetection  
  spanBuildRequestEnd  
Finalize stack step  
  spanRetryLoop  
  InterceptBeforeRetryLoop  
  Retry  
  InterceptAttempt  
  RetryMetricsHeader  
  ResolveAuthScheme  
  GetIdentity  
  ResolveEndpointV2  
  disableHTTPS  
  ComputePayloadHash  
  setLegacyContextSigningOptions  
  InterceptBeforeSigning  
  Signing  
  InterceptAfterSigning  
Deserialize stack step  
  AddRawResponseToMetadata  
  ErrorCloseResponseBody  
  CloseResponseBody  
  ResponseErrorWrapper  
  RequestIDRetriever
```

InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```

func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {

```

```

return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```


spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {

```

    "type": "integer"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
    "type": "long"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      }
    }
  }
}

```

```

    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if\n    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      }
    },
    "InstanceType": {

```

```

    "target": "smithy.api#String",
    "traits":
{
    "smithy.api#documentation": "<p>The instance type of the resource.</p>"
}
},
"AccountID": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
},
"Status": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
},
"Region": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
},
"UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
},
"ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
},
"LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
}

```



```

    },
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
      }
    },
    "OsVersion": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
      }
    },
    "SubscriptionProviderCreateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
      }
    },
    "SubscriptionProviderUpdateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
      }
    },
    "DualSubscription": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
      }
    },
    "RegisteredWithSubscriptionProvider": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        }
      ]
    }
  }

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {

```

```

        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "Boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
]

```

```

    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ]
  }
],
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",

```

```

        "argv": [
            {
                "ref": "Region"
            }
        ]
    },
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ],
                "assign": "PartitionResult"
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseFIPS"
                            },
                            true
                        ]
                    },
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "ref": "UseDualStack"
                            },
                            true
                        ]
                    }
                ]
            }
        ],
        "rules": [
            {
                "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        }
      ]
    }
  ],
  "supportsFIPS"
}
]
},
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }

```

```

    ],
    "type": "tree"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
}
],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ]
    }
  ]
}

```



```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
}
],
"rules": [
  {

```

```

"conditions": [
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {
            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        }
    ],

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
  }
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```

```

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {

```

```

        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
    
```



```

    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with fips disabled and dualstack enabled",
  "expect": {
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "Missing region",

  "expect": {
    "error": "Invalid Configuration: Missing Region"
  }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
}
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists

```

the running Amazon EC2 instances that were discovered with commercial Linux subscriptions.

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptionInstances"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Instances"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n      <ul>\n        <li>\n          <p>\n            <code>AccountID</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>AmiID</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>DualSubscription</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>InstanceID</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>InstanceType</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>ProductCode</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>Region</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>Status</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>UsageOperation</code>\n          </p>\n        </li>\n      </ul>\n      <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>equals</code>\n          </p>\n        </li>\n        <li>\n          <p>\n            <code>Notequal</code>\n          </p>\n        </li>\n      </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    }
  },
  "NextToken": {
```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
                }
            }
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
            "type": "operation",
            "input": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
            },
            "output": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
            },
            "errors": [
                {
                    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
                }
            ],

```

```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
      "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
      returned results will include data aggregated across your accounts in\n  Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n  more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n  partially match, or don't
match a certain subscription's name.</p>\n  <p>The valid names for this filter are:</p>\n  <ul>\n
<li>\n    <p>\n      <code>Subscription</code>\n    </p>\n  </li>\n  </ul>\n
<p>The valid Operators for this filter are:</p>\n  <ul>\n    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    }
  }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    }
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListRegisteredSubscriptionProviders"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "RegisteredSubscriptionProviders"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\t\tin the list.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/RegisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
          "smithy.api#required": {}
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {

```



```

    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
            }
        },
        "SubscriptionProviderSource": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
            "traits": {
                "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
            }
        },
        "SecretArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
            }
        },
        "SubscriptionProviderStatus": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
            }
        },
        "SubscriptionProviderStatusMessage": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
            }
        },
        "LastSuccessfulDataRetrievalTime": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
            }
        }
    },

```

```

    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",

```

```

        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
        }
    },
    "traits": {}
},

```

```

    "Type": {
      "target": "smithy.api#String",
      "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {

```



```

    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        }
      ]
    }
  }

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": true
        }
      }
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated"
    }
  }
}

```

```

from.</p>",
    "smithy.api#required": {}
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.55 busybox 1.37.0-r13

1.55.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source
of bzip2 version 1.0.4. bzip2 source is distributed
under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2006 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must
not be misrepresented
as being the original software.
4. The name of the author may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their

rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the

notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and
copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.
- c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution
of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is

void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE

PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice

like this

when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it

under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of

Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.56 musl 1.2.5-r9

1.56.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

Copyright 2005-2020 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Authors/contributors include:

A. Wilcox
Ada Worcester
Alex Dowad
Alex Suykov
Alexander Monakov
Andre McCurdy
Andrew Kelley
Anthony G. Basile
Aric Belsito
Arvid Picciani
Bartosz Brachaczek
Benjamin Peterson
Bobby Bingham
Boris Brezillon
Brent Cook
Chris Spiegel
Clment Vasseur
Daniel Micay
Daniel Sabogal
Daurnimator
David Carlier
David Edelsohn
Denys Vlasenko
Dmitry Ivanov
Dmitry V. Levin
Drew DeVault
Emil Renner Berthing
Fangrui Song
Felix Fietkau
Felix Janda
Gianluca Anzolin
Hauke Mehrrens
He X

Hiltjo Posthuma
Isaac Dunham
Jaydeep Patil
Jens Gustedt
Jeremy Huntwork
Jo-Philipp Wich
Joakim Sindholt
John Spencer
Julien Ramseier
Justin Cormack
Kaarle Ritvanen
Khem Raj
Kylie McClain
Leah Neukirchen
Luca Barbato
Luka
Perkov
M Farkas-Dyck (Strake)
Mahesh Bodapati
Markus Wichmann
Masanori Ogino
Michael Clark
Michael Forney
Mikhail Kremnyov
Natanael Copa
Nicholas J. Kain
orc
Pascal Cuoq
Patrick Oppenlander
Petr Hosek
Petr Skocik
Pierre Carrier
Reini Urban
Rich Felker
Richard Pennington
Ryan Fairfax
Samuel Holland
Segev Finer
Shiz
sin
Solar Designer
Stefan Kristiansson
Stefan O'Rear
Szabolcs Nagy
Timo Ters
Trutz Behn
Valentin Ochs
Will Dietz

William Haddon
William Pitcock

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier or Copyright 2017-2018 Arm Limited and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (`src/string/arm/memcpy.S`) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The AArch64 memcpy and memset code (`src/string/aarch64/*`) are Copyright 1999-2019, Arm Limited.

The implementation of DES for crypt (`src/crypt/crypt_des.c`) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (`src/crypt/crypt_blowfish.c`) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (`src/stdlib/qsort.c`) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the ellcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream

integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed

above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/* and arch/*/bits/*) and crt files intended to be linked into applications (crt/*, ldso/dlstart.c, and arch/*/crt_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham
John Spencer
Nicholas J. Kain
Rich Felker
Richard Pennington
Stefan Kristiansson
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated

the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

1.57 pax-utils 1.3.8-r1

1.57.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and

distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies
a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.58 openssl 3.3.5-r0

1.58.1 Available under license :

```
# Contributor: Patrycja Rosa <alpine@ptrcnll.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnll.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"
builddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
```

```
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"
```

Original Copyright (c) 1985 by Supoj Sutanthavibul

Parts Copyright (c) 1989-2012 by Brian V. Smith

Parts Copyright (c) 1991 by Paul King

Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)

The text rotation code in w_rottext.c was written by Alan Richardson.

The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker

The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick

The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.22

pkgrel=1

pkgdesc="Various data formats for the SPDX License List"

url="https://spdx.org/"

arch="noarch"

options="!check" # no test suite

license="CC-BY-3.0"

```

source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdft rdfturtle rdfxml template text"

for type in $_types; do
    subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
    mkdir -p "$pkgdir"
}

_subpkg() {
    local type=${subpkgname/$pkgname-/}
    pkgdesc="$pkgdesc ($type)"
    install_if="$pkgname=$pkgver-r$pkgrel"
    mkdir -p "$subpkgdir"/usr/share/spdx
    cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}

list() {
    pkgdesc="$pkgdesc (licence list)"
    mkdir -p "$subpkgdir"/usr/share/spdx
    local i; for i in "$builddir"/text/*.txt; do
        local
        license=${i##*/}
        echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
    done
}

sha512sums="
b213fe66699770d75a4c994a01a5c08325751423516c2fb871088a47e7e7ce605736064f5ce3a63dc1d3c462271832db
7bc464ac968df1888f9823787964c786 license-list-data-3.22.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=1
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"

```

```
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz"
options="!check" # no tests
```

```
build() {
# some GNU autoconf options are ignored, but it works
./configure \
--build=$CBUILD \
--host=$CHOST \
--prefix=/usr \
--sysconfdir=/etc \
--mandir=/usr/share/man \
--localstatedir=/var/lib # it really shouldn't be in /var
make
}
```

```
package() {
make DESTDIR="$pkgdir" install
install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}
```

```
sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
bsd-games-3.3.tar.gz
"
```

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in

exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

- a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,
- b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,
- c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that

accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY

YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ``Gnomovision'` (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.

- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 - d) make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
 - a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names,
and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Contributor: Dhruvin Gandhi <contact@dhruvin.dev>

Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

_pkgname=license-expression

pkgver=30.3.0

pkgrel=1

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools_scm py3-gp517 py3-wheel"

```

checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir/${_pkgname}-${pkgver}"

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

check() {
    python3 -m venv --clear --without-pip --system-site-packages .testenv
    .testenv/bin/python3 -m installer .dist/*.whl
    .testenv/bin/python3 -m pytest
}

package() {
    python3 -m installer -d "$pkgdir" \
        .dist/*.whl
}

sha512sums="
3fe4ac8830085fe38d8ee713a332047941b2a7a8ff5f31810b124c5903335a8343a222926aefe000da66a49b4d6f8b934
af3cc350448b075a4ec6aa1fdb51aef
    license-expression-30.3.0.tar.gz
"

```

Apache License
 Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

```
# Automatically generated by apkgbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=9
#_pkgreal is used by apkgbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
buildid="$srcdir/$_pkgreal-$pkgver"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
```

```

export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

Vintage Dreams Waves v 2.0. for Creative Labs' AWE Soundcards
(EMU Soundfont 2 Format)

```

Copyright (c) Ian Wilson, 1996 (Updated January 1998)

This soundfont is freeware. You may freely use and/or redistribute it subject to the following terms:

1. It is not altered, edited, modified, ripped, or converted to other formats, except for private use only.
2. It is distributed with this copyright notice.

This soundfont is distributed WITHOUT WARRANTY, and without the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. No liability or damages can be inferred upon the said copyright owner, Ian Wilson.

Any feedback, contact Ian Wilson.

vintagedreamworks@hotmail.com

<http://www.geocities.com/SiliconValley/Campus/8645/index.html>

<http://members.nbci.com/silicon39/>

<http://www.mp3.com/silicon39>

<https://analoguesque.x10host.com/>

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

mini_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=libart-lgpl

pkgver=2.3.21

pkgrel=8

pkgdesc="A library for high-performance 2D graphics"

url="https://www.levien.com/libart/"

arch="all"

options="!check" # No test suite.

license="LGPL-2.0-or-later"

subpackages="\$pkgname-dev"

source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-\$pkgver.tar.bz2"

"

builddir="\$srcdir"/libart-lgpl-\$pkgver

```

prepare() {
    update_config_sub
    default_prepare
}

build() {
    ./configure \
    --build=$CBUILD \
    --host=$CHOST \
    --prefix=/usr
    make
}

package() {
    make DESTDIR="$pkgdir" install
}

```

```

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"

```

1.59 alpine-base 3.21.5-r0

1.59.1 Available under license :

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.60 procs 0.19.1

1.60.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

procfs provides functions to retrieve system, kernel and process metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

1.61 tre 0.8.0-r2

1.61.1 Available under license :

This is the license, copyright notice, and disclaimer for TRE, a regex matching package (library and tools) with support for approximate matching.

Copyright (c) 2001-2009 Ville Laurikari <vl@iki.fi>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.62 aws-sdk-go-v2-internal-configsources

1.4.11

1.62.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step

```

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,

```

```

c.addOperationCreateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //

```

```

// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
    if err

```

```

= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

```

```
}
CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
```

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

```

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU

```

```

//
// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},

```

```

middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

```

DeleteLicense

```

```

Initialize stack step

```

```

spanInitializeStart

```

```

InterceptExecution

```

```

RegisterServiceMetadata

```

```

legacyEndpointContextSetter

```

```

SetLogger

```

```

OperationInputValidation

```

```

spanInitializeEnd

```

```

Serialize stack step

```

```
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licensemanager
```

```
import (
    "context"
```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

```

```

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}

```

```

}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains

```


details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"

```
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
```

"smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"

```
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
```

```
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
    },
```

```
    "DomainIpv4List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
```

```
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
```

```
    "DomainIpv6List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
```

```

Directory.</p>",
    "smithy.api#length": {
        "min": 1,
        "max": 2
    }
},
"DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
},
"DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "SELF_MANAGED",
                "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-

```

```

zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      },
    },
  },

```

```

    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
  {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {

```

```

    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {

```



```

    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid"
      }
    }
  }
}

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^([a-z0-9-]{10})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",

```

```

    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
        "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }

```



```

    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    },
    "Products": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
}

```

```

},
"com.amazonaws.licenseanagerusersubscriptions#IpV4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IpV4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IpV6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
    }
  ],

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*"
    ]
  }
}

```

```

        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",

```

```

        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    }
]

```

```

        ]
    }
},
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
},
"type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [

```



```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
            ],
        },
    ],
    true
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
    ],
    true
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                        ],
                        "supportsFIPS"
                    }
                ],
            }
        ],
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
},
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            },
            {
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    }
                ]
            }
        ]
    }

```

```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
},
],
"rules": [
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"

```

```

        },
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
    },

```

```

        "params": {
            "Region": "us-east-1",

"UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,

            "UseDualStack": true
        }
    }

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},

```



```

{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint":
{
      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```

```

    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    }
  }
}

```

```

    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
          }
        },
        },
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>"
          }
        },
        },
        "ServerEndpoint": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
          "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
          }
        },
        },
        "StatusMessage": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
          }
        },
        },
        "LicenseServerEndpointId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
          "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {

```

```

"type": "string",
"traits": {
  "smithy.api#enum": [
    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {

```

```

        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
    },
    {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single

```



```

request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstances": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/instance/ListInstances"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceSummaries"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {

```

```

    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
      "InstanceSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoints": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits":

```

```

{
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
}
},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummaries": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
        }
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
    ]

```



```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
        "method": "GET",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",

```

```

        "smithy.api#required": {}
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {

```

```

"type": "structure",
"members": {
  "RdsSalCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>The CredentialsProvider resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
      "smithy.api#required": {}
    }
  },
},
"traits": {

  "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],

```

```

"traits": {
  "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/RegisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {

        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",

```



```

        "smithy.api#length": {
            "min": 1
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-(([0-9a-z]{8})|([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services

```

Billing. For more information, see <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html> Viewing your monthly charges in the Amazon Web Services Billing User Guide.

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",

"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
          }
        },
        "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want

```



```

to tag.</p>",
    "smithy.api#httpLabel": {},
    "smithy.api#required": {}
  }
},
"Tags": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

```

"code": 200,

```

    "method": "POST",
    "uri": "/identity-provider/UpdateIdentityProviderSettings"

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
      }
    },
    "UpdateSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
      "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",

```

```

        "traits": {
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    }
}

```

```

    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //

```

```

The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {

```



```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

}

out := result.(*GetLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);

```

```

err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:

```

```

python-version: [3.9]

steps:
  - name: Checkout target
    uses: actions/checkout@v2
    with:
      path: sdkbase
      ref: ${ github.base_ref }
  - name: Checkout this ref
    uses: actions/checkout@v2
    with:
      path: new-ref
      fetch-depth: 0
  - name: Get Diff
    run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
  - name: Get Target Files
    run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
  - name: Checkout scancode
    uses: actions/checkout@v2
    with:
      repository: nexB/scancode-toolkit
      path: scancode-toolkit
      fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

```



```

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

```

```

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}

```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{

```

```

    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
  }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Checks out the specified license for offline use.
```

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

```

```

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

```

```

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}

```

```

}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```



```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "AssociateLicense",
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

```

```

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```



```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {

```

```

return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
return err
}
if err = addInterceptExecution(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
return err
}
if err = addInterceptTransmit(stack, options); err != nil {
return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckInLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
}

```

```

if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}

```


CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {

```

```

    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {

```



```

    },
    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    }
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
  "type": "structure",
  "members": {}
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see<n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage<n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage<n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```



```

    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",

```

```

"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    }

    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

    ],
    "type": "tree"
}

    ],

```

```

    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"

```

```

        },
        true
    ]
}

],

"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        }
                    ]
                }
            ],
            "supportsFIPS"
        }
    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
},
{
    "conditions": [],
}
],

"rules": [
{

```

```

        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

```



```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "error": "DualStack is enabled but this partition does not support DualStack",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        }
    ]
}

```

```

    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  }
}

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```

```

    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
      }
    }
  }
},
{
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
},
{
  "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
}

```



```

    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },

```

```

        {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        }
    ]
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
        <ul>\n
            <li>\n                <p>\n                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \t\define the behavior of the filter:</p>\n
        <ul>\n
            <li>\n                <p>\n                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n                <p>\n                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
},

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
        "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
    returned results will include data aggregated across your accounts in\n  Organizations.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptions"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Subscriptions"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
        that match one or\n  more sets of key-value pairs that you specify. For example, you can filter by the name of\n
        <code>Subscription</code> with an optional operator to see subscriptions that match,\n  partially match, or don't
        match a certain subscription's name.</p>\n  <p>The valid names for this filter are:</p>\n  <ul>\n
        <li>\n  <p>\n  <code>Subscription</code>\n  </p>\n  </li>\n  </ul>\n
        <p>The valid Operators for this filter are:</p>\n  <ul>\n  <li>\n  <p>\n  <code>contains</code>\n  </p>\n  </li>\n  <li>\n  <p>\n  <code>equals</code>\n  </p>\n  </li>\n  <li>\n  <p>\n  </li>\n  </ul>\n
      "
    }
  }
}

```

```

<code>Notequal</code>\n          </p>\n          </li>\n        </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",

```

```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSources": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
        }
      },
      "MaxResults": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {

```

```

"name": "Disabled",
  "value": "Disabled",
  "documentation": "Disabled OrganizationIntegration"
}
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {

```



```

subscription provider \n\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      }
    },
  },

```

```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",

```

```

        "documentation": "InProgress status"
    },
    {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [

```



```

        {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
    },

```


"smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are formatted as key value pairs.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
  "smithy.api#http": {
    "method": "DELETE",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",

  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tagKeys": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input":
    {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",

```

```

        "smithy.api#required":
    {}
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    }
}

```

```

    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

InterceptAfterDeserialization

OperationDeserializer

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.63 mime-type 1.4.10

1.63.1 Available under license :

Copyright (c) 2013-present Contentful GmbH

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT

1.64 prometheus-client-model 0.6.2

1.64.1 Available under license :

Data model artifacts for Prometheus.

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.65 libmd 1.1.0-r0

1.65.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:

*

Copyright:

Copyright 2009, 2011, 2016 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/sha2.h

src/sha2.c

Copyright:

Copyright 2000-2001, Aaron D. Gifford

All rights reserved.

License: BSD-3-clause-Aaron-D-Gifford

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTOR(S) ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/rmd160.h

src/rmd160.c

Copyright:

Copyright 2001 Markus Friedl. All rights reserved.

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE

FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

src/md2.c

Copyright:

Copyright (c) 2001 The NetBSD Foundation, Inc.

All rights reserved.

.

This code is derived from software contributed to The NetBSD Foundation
by Andrew Brown.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must
reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

Files:

man/rmd160.3

man/sha1.3

man/sha2.3

Copyright:

Copyright 1997, 2003, 2004 Todd C. Miller <Todd.Miller@courtesan.com>

License: ISC

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above

copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

man/mdX.3

src/helper.c

Copyright:

Poul-Henning Kamp <phk@login.dkuug.dk>

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this file. As long as you retain
this notice you

can do whatever you want with this stuff. If we meet some day, and you think
this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

Files:

include/md4.h

src/md4.c

Copyright:

Colin Plumb

Todd C. Miller

License: public-domain-md4

This code implements the MD4 message-digest algorithm.

The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.

This code is in the public domain; do with it what you wish.

Todd C. Miller modified the MD5 code to do MD4 based on RFC 1186.

Files:

include/md5.h

src/md5.c

Copyright:

Colin Plumb

License: public-domain-md5

This code implements the MD5 message-digest algorithm.

The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.

This code is in the public domain; do with it what you wish.

Files:

include/sha1.h

src/sha1.c
Copyright:
Steve Reid <steve@edmweb.com>
License: public-domain-sha1
100% Public
Domain

1.66 prometheus-common 0.67.1

1.66.1 Available under license :

Common libraries shared by Prometheus Go components.
Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.67 aws-sdk-go-v2-service-internal-s3shared

1.19.11

1.67.1 Available under license :

- GetLicense
- Initialize stack step
- spanInitializeStart
- InterceptExecution
- RegisterServiceMetadata
- legacyEndpointContextSetter
- SetLogger
- OperationInputValidation

```
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

```

```

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

```

```

type CreateLicenseOutput struct {

    // Amazon Resource Name (ARN) of the license.
    LicenseArn *string

    // License status.
    Status types.LicenseStatus

    // License version.
    Version *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```



```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
    spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
    setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
RecursionDetection
    spanBuildRequestEnd
Finalize stack step
    spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader

```

ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection

```

spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

```

```

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string
}

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```

```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```



```

if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {

```

```
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
```

DeleteLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU

```

```

//
// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
}

```

```

if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}
DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step

```

ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 },
 "traits": {
 "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
 "smithy.api#error": "client"
 }
 }
 }


```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
    "type": "structure",
    "members": {
      "DirectoryId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
        "traits": {
          "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
      "type": "structure",
      "members": {
        "DomainName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
          }
        }
      }
    }
  }
}

```

```

    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {

```

```

        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
    },
    {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
    }
]
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {

```

```

        "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    },
    "traits": {
        "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/AssociateUser"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>The identity provider for the user.</p>",
                "smithy.api#required": {}
            }
        },
        "Domain": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
    "smithy.api#error": "server"
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/license-server/CreateLicenseServerEndpoint"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members":

```

```

{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>"
            }
        }
    }
}

```

resource that was deleted.</p>"

```
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      }
    ],
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#Directory": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
    "type": "operation",
    "input": {

      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {

        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",

```

```

        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
        },
        "InstanceUserArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
            }
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {

```

```

    "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
  }
},
"Value": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>Value of the filter.</p>"
  }
}
},
"traits": {
  "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
  "type": "union",
  "members": {
    "ActiveDirectoryIdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
  }
}

```



```

    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
          "smithy.api#required": {}
        }
      },
      "Products": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
          "smithy.api#required": {}
        }
      },
      "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
      },
      "OwnerAccountId": {
        "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    },
    "DisassociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {

```

```

    "type": "list",
    "member": {

"target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}):[0-9a-fA-F]{1,4})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
},

```

```

{
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
    "aws.api#service": {
        "sdkId": "License Manager User Subscriptions",
        "arnNamespace": "license-manager-user-subscriptions",
        "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
        "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
        "additionalAllowedHeaders": [
            "*",
            "content-type",
            "x-amz-content-sha256",
            "x-amz-user-agent",
            "x-amzn-platform-id",
            "x-amzn-trace-id",
            "content-length",
            "x-api-key",
            "authorization",
            "x-amz-date",
            "x-amz-security-token",
            "Access-Control-Allow-Headers",
            "Access-Control-Allow-Methods",
            "Access-Control-Allow-Origin"
        ],
        "additionalExposedHeaders": [
            "x-amzn-errortype",
            "x-amzn-requestid",
            "x-amzn-trace-id"
        ]
    },
    "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager User Subscriptions",
    "smithy.rules#endpointRuleSet": {
        "version": "1.0",
        "parameters": {
            "Region": {
                "builtIn": "AWS::Region",
                "required": false,

```

```

        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
]
},
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    }
                ]
            }
        ]
    }
]

```

```

        },
        true
    ]
}
],
"error": "Invalid Configuration: FIPS and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [

{
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    }
],
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        },

    "properties": {},
    "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
}

```



```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsFIPS"
        ]
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}

```

```

    ],
    "type": "tree"
  },
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "fn": "getAttr",
                "argv": [
                  {
                    "ref": "PartitionResult"
                  },
                  "supportsFIPS"
                ]
              },
              true
            ]
          }
        ],
        "rules": [
          {
            "conditions": [],
            "endpoint": {
              "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
              "properties": {},
              "headers": {}
            },
            "type": "endpoint"
          }
        ],

```

```

        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS is enabled but this partition does not support FIPS",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {

```

```

        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

```

```

"expect": {
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    }
},
"params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": true
}
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

```

```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS":
false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",

```



```

        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint":
{
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",

```

```

    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
      }
    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },

```

```

        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips disabled and dualstack enabled",
        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            },
            "Ipv4Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
                },
            },
            "Ipv6Address": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
                },
            },
            "traits": {
                "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
            },
        },
        "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
            "type": "structure",
            "members": {
                "IdentityProviderArn": {
                    "target": "smithy.api#String",
                    "traits": {
                        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                    },
                },
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                },
            },
        },
    },

```

```

    "ServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
      }
    ]
  }
}

```

```

        "name": "DELETED",

"value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,

```



```

        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request

```

```

criteria.</p>",
    "smithy.api#required": {}
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ]
},

```

```

"traits": {
  "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/instance/ListInstances"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

```

```

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      }
    ]
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  }
}

```

```

    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a product for this user.</p>",
      "smithy.api#required": {}
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
    }
  },
  "SubscriptionStartDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The start date of a subscription.</p>"
    }
  },
  "SubscriptionEndDate": {
    "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
        "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      }
    }
  }

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",

```

```

"members": {
  "Subnets": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
      "smithy.api#length": {
        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },

    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```



```

    ],
    "traits": {
      "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
  },

```

```

    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
      },
    },
  },

```

```

    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related

```

configuration settings. You can update any combination of settings in a single operation such as the:

- Subnets which you want to add to provision VPC endpoints.
- Subnets which you want to remove the VPC endpoints from.
- Security group ID which permits traffic to the VPC endpoints.

```

    "smithy.api#required": {}
  }

  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    },

    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Removes the Grafana Enterprise license from a workspace.

```

func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)

```

```

if err != nil {
    return nil, err
}

out := result.(*DisassociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```



```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```

```

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}

```

```

}
}
name: License Scan

on: [pull_request]

permissions:
  contents: read

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${{ github.base_ref }}
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${{ matrix.python-version }}
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit

```

```

run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
  toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
  are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
    c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

//

```



```

Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}

```

```

}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

}

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```



```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package grafana

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more

```

```

// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput

```

```

struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
}

```

```

if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptExecution(stack,
options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
return err
}
if err = addInterceptTransmit(stack, options); err != nil {
return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
}

```

```

if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.

```

```

//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {

```



```

return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
}

```

```

if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {

```

```

return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DeleteLicense",
}
}

```

```

CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```

UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
c.addOperationCheckInLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckInLicenseInput struct {
```

```
    // License consumption token.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseConsumptionToken *string
```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```



```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

```

```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ]
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    }
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        }
    },
    "traits": {}
},

```

```

    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
"list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],

```

```

    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation":
            "<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
          "smithy.api#required": {}
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified in the request.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {

```

```
"smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the last \n\t\t\t\t\tsuccessful subscription data request.</p>"
      }
    },
    "SubscriptionProviderStatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The detailed message from your subscription provider token status.</p>"
      }
    },
    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved subscription details \n\t\t\t\t\tfrom your registered third-party Linux subscription provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
      },
      "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
          "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
      }
    },
  },

```

```

    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      }
    }
  }
}

```



```
"traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
},
"DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for\n\t\t\tthe same software on your instance.</p>"
    }
},
"RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta third-party Linux subscription provider that you've registered with License Manager.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
```

```

"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*"
    ]
  }
}

```

```

        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    }
},
"rules": [

```

```

{
  "conditions": [
    {
      "fn": "isSet",
      "argv": [
        {
          "ref": "Endpoint"
        }
      ]
    }
  ]
},
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ]
  }
],
"error": "Invalid Configuration: FIPS and custom endpoint are not supported",
"type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{

```

```

        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [

```

```

    {
      "fn": "booleanEquals",
      "argv": [
        {

"ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"

                },

                "supportsFIPS"
              ]
            }
          ]
        },
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}
]
}
],
"rules": [
    {
        "conditions": [],

"rules": [

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {

"conditions": [

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"

```



```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {

"fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    }
},
],
"rules": [
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

"properties": {},

                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [],
        "rules": [
            {
                "conditions": [],

```

```

        "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",

```

```

        "type": "error"
      }
    ],
    "type": "tree"
  }
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    }
  ]
}

```

```

    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    }
  },
  {
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    }
  },
  {
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
}

```

```

    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"

```

```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  }
},
{
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
}

```



```

    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
}

```

```

        }
    ],
    "version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto

```

```

streamline results:
<ul>
  <li>
    <p>
      <code>AccountID</code>
    </p>
  </li>
  <li>
    <p>
      <code>AmiID</code>
    </p>
  </li>
  <li>
    <p>
      <code>DualSubscription</code>
    </p>
  </li>
  <li>
    <p>
      <code>InstanceID</code>
    </p>
  </li>
  <li>
    <p>
      <code>InstanceType</code>
    </p>
  </li>
  <li>
    <p>
      <code>ProductCode</code>
    </p>
  </li>
  <li>
    <p>
      <code>Region</code>
    </p>
  </li>
  <li>
    <p>
      <code>Status</code>
    </p>
  </li>
  <li>
    <p>
      <code>UsageOperation</code>
    </p>
  </li>
</ul>
<p>For each filter, you can use one of the following <code>Operator</code> values to define the behavior of the filter:
<ul>
  <li>
    <p>
      <code>contains</code>
    </p>
  </li>
  <li>
    <p>
      <code>equals</code>
    </p>
  </li>
  <li>
    <p>
      <code>Notequal</code>
    </p>
  </li>
</ul>
}
},
"MaxResults": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
  "traits": {
    "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
  }
},
"NextToken": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from a previously truncated response.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 16384
    }
  }
},
"traits": {
  "smithy.api#documentation":
    "NextToken length limit is half of ddb accepted limit. Increase this limit if parameters in request increases."
},
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
  "type": "structure",
  "members": {
    "Instances": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
      }
    }
  },
  "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":
            "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {

```

```

    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n
<code>contains</code>\n        </p>\n        </li>\n        <li>\n            <p>\n
<code>equals</code>\n        </p>\n        </li>\n        <li>\n            <p>\n
<code>NotEqual</code>\n        </p>\n        </li>\n    </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            }
        }
    }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},

```

"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {

```

  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],

```

```

    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    }
  }

```



```

        "smithy.api#httpLabel": {},
        "smithy.api#required": {}

    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring

```



```

    }
  },
  "SubscriptionProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    }
  }
}

```

```

    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {

```

"smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

```

    }
  },
  "InstanceCount": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
    "traits": {
      "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        },
        {
          "name": "PENDING",
          "value": "PENDING",
          "documentation": "PENDING status"
        }
      ]
    }
  }
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
  {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
    "type": "structure",

```



```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

```



```

        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
            {}
        }
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
            "smithy.api#required": {}
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        }
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }

    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.68 confluentinc-confluent-kafka-go 2.12.0

1.68.1 Available under license :

LICENSE

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill
2023, Confluent Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

For src/crc32c.c copied (with modifications) from
<http://stackoverflow.com/a/17646775/1821055>

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
This software is provided 'as-is',

without any express or implied warranty. In no event will the author be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler
madler@alumni.caltech.edu

*/

LICENSE.fnv1a

parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please
do not copyright this code. This code is in the public domain.

LANDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LANDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

By:
chongo <Landon Curt Noll> /oo/
<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram>

The MIT License
(MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

LICENSE.lz4

src/rdxxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library
Copyright (c) 2011-2020, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.murmur2

parts of src/rdmurmur2.c: [git@github.com:abandoned/murmur2.git](https://github.com:abandoned/murmur2.git)

MurMurHash2 Library

//-----
// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.nanopb

For
files in src/nanopb :
<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.opentelemetry

For files in src/opentelemetry: <https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.
=====

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted,
free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

LICENSE.queue

For
sys/queue.h:

- * Copyright (c) 1991, 1993
- * The Regents of the University of California. All rights reserved.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 4. Neither the name of the University nor the names of its contributors
- * may be used to endorse or promote products derived from this software
- * without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND

* ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* \$FreeBSD\$

LICENSE.regexp

regexp.c and regexp.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

LICENSE.snappy

LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h #
originally retrieved from <http://github.com/andikleen/snappy-c> #
git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #
#####

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions
in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the

distribution.

* Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.tinychread

From <https://github.com/tinychread/tinychread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

/*

* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>

*

* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.

*

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
* ANY DAMAGES

* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

*

* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.

*/

/*-

* Copyright (c) 2000 The NetBSD Foundation, Inc.

* All rights reserved.

*

* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.

*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

- * 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
* notice,

this list of conditions and the following disclaimer in the

* documentation and/or other materials provided with the distribution.

*

* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.

*/

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You"
(or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work

or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.69 aws-sdk-go-v2-service-s3 1.88.7

1.69.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step


```

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

```

```

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.

```

Version *string

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde

}

```
func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
```

```
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
```

```
    if err != nil {
```

```
        return err
    }
```

```
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
```

```
    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
```

```
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
```

```
    if err = addClientRequestID(stack); err != nil {
        return err
    }
```

```
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
```

```
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
```

```
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
```

```
    if err = addRetry(stack, options); err != nil {
        return err
    }
```

```
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}

```

```

}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil

```

```

}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CreateLicense",
}
}
CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning

```

Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions

InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity

```

ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Lists received licenses.
```

```
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
```

```

    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata

```

```

    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart

```

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger

```

InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

```



```

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {

```

```

    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}

```

```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {

```

```

return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "ListLicenses",
}
}
DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

```

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "ActiveDirectorySettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"DomainIpv6List": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
  "traits": {
    "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
    "smithy.api#length": {
      "min": 1,
      "max": 2
    }
  }
},
"DomainCredentialsProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
  "traits": {
    "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
  }
},
"DomainNetworkSettings": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
  "traits": {
    "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "SELF_MANAGED",
        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
      },
      {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
      }
    ]
  }
}

```

```

    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a

```


href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly charges in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",

```
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
      }
    }
  }
}
```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },

```

```

    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
      "type": "union",
      "members": {
        "SecretsManagerCredentialsProvider": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
          "traits": {

```

```

        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
    }
}
},
"traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies

```

details for the Active Directory identity provider.</p>"

```
    }
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
```

```
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
```



```

"input": {

"target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {

"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
    },
    "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                    "smithy.api#required": {}
                }
            }
        }
    },

```

```

"traits": {
  "smithy.api#output": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      },
      "Value": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>Value of the filter.</p>"
        }
      }
    },
    "traits": {

```

"smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the registered identity providers product related configuration settings, such as the subnets to provision VPC endpoints.</p>",
      "smithy.api#required": {}
    }
  },
}
```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    },
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  }
}

```



```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server",
    "smithy.api#httpError": 500
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4}:{1,7}|([0-9a-fA-F]{1,4}:{1,6}:[0-9a-fA-F]{1,4})$)"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider"
    }
  ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DisassociateUser"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
  },

```

```

"aws.auth#sigv4": {
  "name": "license-manager-user-subscriptions"
},
"aws.protocols#restJson1": {},
"smithy.api#cors": {
  "additionalAllowedHeaders": [
    "*",
    "content-type",
    "x-amz-content-sha256",
    "x-amz-user-agent",
    "x-amzn-platform-id",
    "x-amzn-trace-id",
    "content-length",
    "x-api-key",
    "authorization",
    "x-amz-date",
    "x-amz-security-token",
    "Access-Control-Allow-Headers",
    "Access-Control-Allow-Methods",
    "Access-Control-Allow-Origin"
  ],
  "additionalExposedHeaders": [
    "x-amzn-errortype",
    "x-amzn-requestid",
    "x-amzn-trace-id"
  ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {

"Region": {
  "builtIn": "AWS::Region",
  "required": false,
  "documentation": "The AWS region used to dispatch the request.",
  "type": "string"
},
"UseDualStack": {
  "builtIn": "AWS::UseDualStack",
  "required": true,
  "default": false,
  "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
  "type": "boolean"
},

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ]
},
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [

```

```

{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
        true
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        }
    },
    "properties": {},
    "headers": {}
},
{
    "type": "endpoint"
},
{
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
{
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                ],
            },
        ],
        true
    },
    {
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ],
    },
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                        ],
                        "supportsFIPS"
                    },
                ],
            },
        ],
    },
],
},

```

```

        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
    },
    ],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

```

```

{
    "ref": "UseFIPS"
    },
    true
]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                    "ref": "PartitionResult"
                },
                "supportsFIPS"
            ]
        },
        true
    ]
    }
],
"rules": [
    {
        "conditions": [],
        "endpoint": {

"url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}
],

```



```

        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            true,
                            {
                                "fn": "getAttr",
                                "argv": [
                                    {
                                        "ref": "PartitionResult"
                                    },
                                    "supportsDualStack"
                                ]
                            }
                        ]
                    }
                ],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ]
            }
        ]
    }
]

```

```

        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "DualStack is enabled but this partition does not support DualStack",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
    "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":

```

```

false,
    "UseDualStack": false
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":
false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,

```

```

        "UseDualStack": false
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            },
            "params": {
                "Region": "us-isob-east-1",
                "UseFIPS": false,
                "UseDualStack": false
            }
        },
        {
            "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://example.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": false,
                    "UseDualStack": false,
                    "Endpoint": "https://example.com"
                }
            },
            {
                "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
                "expect": {
                    "endpoint": {
                        "url": "https://example.com"
                    },
                    "params": {
                        "UseFIPS": false,
                        "UseDualStack": false,
                        "Endpoint": "https://example.com"
                    }
                },
                {
                    "documentation": "For custom endpoint with fips enabled and dualstack disabled",

```

```

    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":

```



```

"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
  "traits": {
    "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
  },
  "Ipv4Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
    }
  },
  "Ipv6Address": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      }
    }
  },

```

```

    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
      }
    },
    "LicenseServerEndpointProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
      }
    },
    "LicenseServers": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
      }
    },
    "CreationTime": {
      "target": "smithy.api#Timestamp",

      "traits": {
        "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId": {
      "type": "string"
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "PROVISIONING",
            "value": "PROVISIONING",
            "documentation": ""
          },
          {
            "name": "PROVISIONING_FAILED",
            "value": "PROVISIONING_FAILED",
            "documentation": ""
          },
          {
            "name": "PROVISIONED",
            "value": "PROVISIONED",
            "documentation": ""
          },
          {
            "name": "DELETING",
            "value": "DELETING",
            "documentation": ""
          },
          {
            "name": "DELETION_FAILED",
            "value": "DELETION_FAILED",
            "documentation": ""
          },
          {
            "name": "DELETED",
            "value": "DELETED",
            "documentation": ""
          }
        ]
      }
    }
  }
}

```

```

    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "HEALTHY",
            "value": "HEALTHY",
            "documentation": ""
          },
          {
            "name": "UNHEALTHY",
            "value": "UNHEALTHY",
            "documentation": ""
          },
          {
            "name": "NOT_APPLICABLE",
            "value": "NOT_APPLICABLE",
            "documentation": ""
          }
        ]
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
      "type": "structure",
      "members": {
        "ServerType": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
          "traits": {
            "smithy.api#documentation": "<p>The type of license server.</p>",
            "smithy.api#required": {}
          }
        },
        "ServerSettings": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
            "smithy.api#required": {}
          }
        }
      }
    }
  }

```

```

    }
    },
    "traits": {
      "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/ListIdentityProviders"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",

```

```

        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
                "smithy.api#required": {}
            }
        },
        "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},

```

```

"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
        "InstanceSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
            }
        }
    }
}

```



```

    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  }
},
"traits": {
  "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
},
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {

```

```

        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
    }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

"type": "structure",
"members": {
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        }
    },
    "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
            "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {

```

```

    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of LicenseServerEndpoint resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    }
  }
}

```

```

    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    },
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
}

},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/ListUserAssociations"
        },
    },

```

```

    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
      }
    }
  }
}

```



```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {

"smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
    },
    "Status": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",
            "smithy.api#required": {}
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {

```

```

    "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
      "RdsSalCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
    }
  }
}

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",

```

```

    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17})$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {

```

```

    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "RDS_SAL",
                    "value": "RDS_SAL",
                    "documentation": ""
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
        "smithy.api#error": "client"
    }
},
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {

```

```

        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```



```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
}
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",

"uri": "/user/StartProductSubscription"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummary": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
            }
        },
        "IdentityProvider":
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
  "type": "string",
  "traits": {

```

```

    "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      }
    }
  }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    }
  }
}

```

```

    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

        },
        "TagKeys": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
            }
        ]
    }
}

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {

"code": 200,
        "method": "POST",
        "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
        "type": "structure",
        "members": {
            "IdentityProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
            },
            "Product": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
                }
            },
            "IdentityProviderArn": {
{
                "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
                }
            },
            "UpdateSettings": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
                "traits": {
                    "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li><p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li><p>Subnets which you want to remove the VPC endpoints from.</p> </li> <li><p>Security group ID which permits traffic to the VPC endpoints.</p> </li> </ul>",
                    "smithy.api#required": {}
                }
            }
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
    }
  }
},

```

```

    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
},
"com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Removes the Grafana Enterprise license from a workspace.

```

func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil

```

```

}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}

```

```

    }
    if err = addInterceptTransmit(stack, options); err != nil {
        return err
    }
    if err = addInterceptBeforeDeserialization(stack, options); err != nil {
        return err
    }
    if err = addInterceptAfterDeserialization(stack, options); err != nil {
        return err
    }
    if err = addSpanInitializeStart(stack); err != nil {
        return err
    }
    if err = addSpanInitializeEnd(stack); err != nil {
        return err
    }
    if
    err = addSpanBuildRequestStart(stack); err != nil {
        return err
    }
    if err = addSpanBuildRequestEnd(stack); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "DisassociateLicense",
    }
}

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.

```

```

func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {

```



```

return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err
!= nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,

```

```

options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

```

```

permissions:
  contents: read

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${ github.base_ref }
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit
        run: ./scancode --help
      - name: Run Scan code on target
        run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
        toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
      - name: Run Scan code on pr ref
        run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-

```

```

toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of

```

```

// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.

```

```

LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }

```

```

}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```



```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}

```

```

    }
    if err = addSpanBuildRequestEnd(stack); err != nil {
        return err
    }
    return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {

```

```

    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)

```

```

    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

```

```

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Allowed license entitlements.
    EntitlementsAllowed
    []types.EntitlementData

    // Date and time at which the license checkout expires.

```

```

Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {

```

```

return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
return err
}
}

```

```

if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {

```

```

    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
// Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {

```

```

if params == nil {
    params = &AssociateLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx,
"AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-grafana-enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

```



```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {

```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region

```

```

string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.

```

```

SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```

```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}

```

```

}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

```

CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop

```


Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks in the specified license. Check in a license when it is no longer in use.
```

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
        c.addOperationCheckInLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckInLicenseInput struct {
```

```
    // License consumption token.  
    //  
    // This member is required.  
    LicenseConsumptionToken *string
```

```
    // License beneficiary.  
    Beneficiary *string
```

```
    noSmithyDocumentSerde  
}
```

```

type
    CheckInLicenseOutput struct {
        // Metadata pertaining to the operation's result.
        ResultMetadata middleware.Metadata

        noSmithyDocumentSerde
    }

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if
    err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }

```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
}

```

```

if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID:
        ServiceID,
    OperationName: "CheckInLicense",
}
}
CreateLicense
Initialize stack step
    spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
    setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
    RecursionDetection
    spanBuildRequestEnd
Finalize stack step
    spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing

```

```

InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ],
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own

```

```

License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        }
    },
    "Values": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
        }
    }
},
"Operator": {

```



```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
        "traits": {
            "smithy.api#documentation": "<p>An operator for filtering results.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
        "type":
    "list",
        "member": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
        },
        "output": {
            "target":
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's registered to your account.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/GetRegisteredSubscriptionProvider"
            }
        }
    }
}

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    },
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {

```



```

        "uri": "/subscription/GetServiceSettings"

    },
    "smithy.api#idempotent": {}
}

},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if n      Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
            }
        },
        "Status": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
            "traits": {
                "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
            }
        },
        "StatusMessage": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
            "traits": {
                "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
            }
        },
        "HomeRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n      subscriptions.</p>"
            }
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      },
      "UsageOperation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-

```

```

subscriptions-usage-operation.html\>Usage\n    operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    },
    "ProductCode": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
        "traits": {
            "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\>Usage\n    operation values</a> in the <i>License Manager User Guide</i> .</p>"
        }
    },
    "LastUpdatedTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
        }
    },
    "SubscriptionName": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
        }
    },
    "OsVersion": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
        }
    },
    "SubscriptionProviderCreateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
        }
    },
    "SubscriptionProviderUpdateTime": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
        }
    },
    "DualSubscription": {
        "target": "smithy.api#String",

```

```
"traits": {  
    "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for  
\n\t\tthe same software on your instance.</p>"  
}  
  
},  
  
"RegisteredWithSubscriptionProvider": {  
    "target": "smithy.api#String",  
    "traits": {  
  
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\tta  
third-party Linux subscription provider that you've registered with License Manager.</p>"  
    }  
}  
  
},  
  
"traits": {  
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux  
subscriptions.</p>"  
}  
  
},  
  
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {  
    "type": "list",  
    "member": {  
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"  
    }  
}  
  
},  
  
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {  
    "type": "structure",  
    "members": {  
        "message": {  
            "target": "smithy.api#String"  
        }  
    },  
    "traits": {  
        "smithy.api#documentation":  
"<p>An exception occurred with the service.</p>",  
        "smithy.api#error": "server"  
    }  
}  
  
},  
  
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {  
    "type": "service",  
    "version": "2018-05-10",  
    "operations": [  
        {  
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"  
        },  
        {  
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"  
        },
```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [

```



```

        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Endpoint"
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
}

    ],
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    }
}

```

```

        },
        "type": "endpoint"
    }
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [

"fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {

"ref": "UseFIPS"
                                },

```

```

        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        },
        {
          "fn": "supportsFIPS"
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ]
}

```

```

    }
  ],
  "rules": [
    {
      "conditions": [],

"rules": [

      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],

    "type": "tree"
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
},
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "rules": [
                    {
                        "conditions": [],
                        "endpoint": {
                            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
                            "properties": {},
                            "headers": {}
                        },
                        "type": "endpoint"
                    }
                ],
                "type": "tree"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
    }
],
"type": "tree"

```



```

        }
    ],
    "type": "tree"
}

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},

```



```

"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
      }
    }
  ]
}

```

```

    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  }
}

```

```

    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"

```

```

    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```

```

    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        }
    },
    {
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {

```

```

        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled LinuxSubscriptionsDiscovery"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings": {
    "type": "structure",
    "members": {
      "SourceRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
          "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
          "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {

```



```

<code>Status</code>\n          </p>\n          </li>\n          <li>\n          <p>\n
<code>UsageOperation</code>\n      </p>\n      </li>\n      </ul>\n      <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \t\define the behavior of the filter:</p>\n      <ul>\n
      <li>\n          <p>\n          <code>contains</code>\n          </p>\n          </li>\n          <li>\n
<p>\n          <code>>equals</code>\n          </p>\n          </li>\n          <li>\n
          <p>\n          <code>NotEqual</code>\n          </p>\n          </li>\n      </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation":
      "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
      }
    }
  }
}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
    returned results will include data aggregated across your accounts in\n  Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those

```

that match one or more sets of key-value pairs that you specify. For example, you can filter by the name of `Subscription` with an optional operator to see subscriptions that match, partially match, or don't match a certain subscription's name.

The valid names for this filter are:

- `Subscription`

The valid Operators for this filter are:

- `contains`
- `equals`
- `NotEqual`

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
}

```

```

    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

```



```

    "smithy.api#input": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "tags": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {

```

```

        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
    },
    {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled OrganizationIntegration"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/RegisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
    }
}

```


the Linux subscription provider resource that you registered.</p>"

```
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\t\tthis is the status of the offline token.</p>"
        }
      }
    },
  },
}
```

```

"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "InProgress",
        "value": "InProgress",
        "documentation": "InProgress status"
      },
      {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
      },
      {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
      },
      {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",

```

```

    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      },
      "InstanceCount": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
        "traits": {
          "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:"
      }
    }
  }
}

```

```

9-\\.[1,510]/[a-z0-9-\\.[1,510}]$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        },
        {
          "name": "PENDING",
          "value": "PENDING",
          "documentation": "PENDING status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {

```



```

        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
    }
},
"tags": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
        "smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",
        "smithy.api#required": {}
    }
}
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
    "type": "map",
    "key":
{
    "target": "smithy.api#String"
},
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    }
},

```



```

    "traits": {
      "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
      "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "tagKeys": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested

```

```

    \n\t\t\tresource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#input":
{}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/UpdateServiceSettings"
        },
        "smithy.api#idempotent": {}
    }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
        "smithy.api#required":
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
        "smithy.api#required": {}
      }
    },
    "AllowUpdate": {
      "target": "smithy.api#Boolean",
      "traits": {
        "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>"
      }
    }
  }
}

```

```

    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.70 aws-sdk-go-v2-feature-dynamodb-attributevalue 1.20.19

1.70.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop

```

InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)

```

```

if err != nil {
    return nil, err
}

out := result.(*CreateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.Entitlement

    // Home Region for the license.
    //
    // This member is required.
    HomeRegion *string

    // License issuer.
    //
    // This member is required.
    Issuer *types.Issuer

    // License name.
    //
    // This member is required.

```

```

LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {

```

```

if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err

```

```

= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

```

```
}
CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
```

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step

```

AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU

```

```

//
// - Status
//
//
// - Fingerprint
//
// - IssuerName
//
// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},

```

```

middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

```

```

DeleteLicense

```

```

Initialize stack step

```

```

spanInitializeStart

```

```

InterceptExecution

```

```

RegisterServiceMetadata

```

```

legacyEndpointContextSetter

```

```

SetLogger

```

```

OperationInputValidation

```

```

spanInitializeEnd

```

```

Serialize stack step

```

```
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licensemanager
```

```
import (
    "context"
```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
[]types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

```

```

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }

```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```

```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}

```

```

}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains

```


details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"

```
    }
  },
  "ActiveDirectoryType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
    "traits": {
      "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active
Directory or an Amazon Web Services Managed Active Directory.</p>"
    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
```

"smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"

```
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings": {
```

```
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
    },
```

```
    "DomainIpv4List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Ipv4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
```

```
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
```

```
    "DomainIpv6List": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Ipv6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
```

```

Directory.</p>",
    "smithy.api#length": {
        "min": 1,
        "max": 2
    }
},
"DomainCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
},
"DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "SELF_MANAGED",
                "value": "SELF_MANAGED",
                "documentation": "SELF_MANAGED type of Active Directory"
            },
            {
                "name": "AWS_MANAGED",
                "value": "AWS_MANAGED",
                "documentation": "AWS_MANAGED type of Active Directory"
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-

```

```

zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/AssociateUser"
      },
    },
  },

```

```

    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
  {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licenseanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    }
  ]
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
    "type": "structure",
    "members":
{
        "IdentityProviderArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
                "smithy.api#required": {}
            }
        },
        "LicenseServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {

```

```

    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {

```



```

    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid"
      }
    }
  }
}

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^([a-z0-9-]{10})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/DisassociateUser"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings": {
    "type": "structure",

```

```

    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Filter": {
      "type": "structure",
      "members": {
        {
          "Attribute": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
          },
          "Operation": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
          },
          "Value": {
            "target": "smithy.api#String",
            "traits": {
              "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
        }
      },
      "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
        "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
      "ActiveDirectoryIdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    }
  },

```

```

    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
  {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }

```



```

    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
          "smithy.api#required": {}
        }
      },
      "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        },
        "Domain": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
            }
        },
        "AssociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
            }
        },
        "DisassociationDate": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
        },
        "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
            "type": "list",
            "member": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
            },
            "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
                "type": "structure",
                "members": {
                    "message": {
                        "target": "smithy.api#String"
                    }
                },
                "traits": {
                    "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
                    "smithy.api#error": "server",
                    "smithy.api#httpError": 500
                }
            }
        }
    }

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV4List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV6": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}:[0-9a-fA-F]{1,4}$"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
        }
      ]
    }
  }

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpoints"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*"
      ],

```

```

        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",
        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",

```

```

        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ]
    }
]

```

```

        ]
    }
},
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
},
"type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [

```



```

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
            ],
        },
    ],
    true
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        },
    ],
    true
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                        ],
                        "supportsFIPS"
                    }
                ],
            }
        ],
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
}

]
}
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsFIPS"
                        ]
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [],
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                    "properties": {},
                    "headers": {}
                },
                "type": "endpoint"
            },
            {
                "type": "tree"
            },
            {
                "conditions": [],
                "error": "FIPS is enabled but this partition does not support FIPS",
                "type": "error"
            }
        ],
        "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    }
                ]
            }
        ]
    }

```

```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"

```

```

        },
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
            }
        },
    },

```

```

        "params": {
            "Region": "us-east-1",

"UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
        "expect": {

"endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
true,

            "UseDualStack": true
        }
    }

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS":
false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },

```



```

{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint":
{
      "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```

```

    },
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with fips disabled and dualstack enabled",
      "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "Missing region",
      "expect": {
        "error": "Invalid Configuration: Missing Region"
      }
    }
  ],
  "version": "1.0"
}

},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
  "type": "structure",
  "members": {
    "ProvisioningStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
      "traits": {
        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
      }
    },
    "HealthStatus": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
      "traits": {
        "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
      }
    },
    "Ipv4Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
      }
    }
  }
}

```

```

    },
    "Ipv6Address": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
      }
    },
    },
    "traits": {
      "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>"
        }
      },
      "ServerEndpoint": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
      },
      "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
      }
    }
  }
}

```

```

    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {

```

```

"type": "string",
"traits": {
  "smithy.api#enum": [
    {
      "name": "PROVISIONING",
      "value": "PROVISIONING",
      "documentation": ""
    },
    {
      "name": "PROVISIONING_FAILED",
      "value": "PROVISIONING_FAILED",
      "documentation": ""
    },
    {
      "name": "PROVISIONED",
      "value": "PROVISIONED",
      "documentation": ""
    },
    {
      "name": "DELETING",
      "value": "DELETING",
      "documentation": ""
    },
    {
      "name": "DELETION_FAILED",
      "value": "DELETION_FAILED",
      "documentation": ""
    },
    {
      "name": "DELETED",
      "value": "DELETED",
      "documentation": ""
    }
  ]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {

```

```

        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
    },
    {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings for your server.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single

```



```

request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListInstances": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/instance/ListInstances"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceSummaries"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {

```

```

    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
    "type": "structure",

    "members": {
      "InstanceSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/ListLicenseServerEndpoints"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "LicenseServerEndpoints"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {

```

```

    "type": "structure",
    "members": {
      "MaxResults": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
          "smithy.api#range": {
            "min": 1,
            "max": 100
          }
        }
      },
      "Filters": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
      "type": "structure",
      "members": {
        "LicenseServerEndpoints": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits":

```

```

{
    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
}
},
"traits": {
    "smithy.api#output": {}
},
"com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity
provider.</p>",

```

```

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
    }
},
"errors": [
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
]

```



```

    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
        "method": "GET",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
        "ResourceArn": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
        "type": "structure",
        "members": {
            "Tags": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
                "traits": {
                    "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
                }
            },
            "traits": {
                "smithy.api#output": {}
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {

```

```

"type": "operation",
"input": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/ListUserAssociations"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceUserSummaries"
  }
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
      "InstanceUserSummaries": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't empty, there are additional elements that the service hasn't included in this request. Use this token with the next request to retrieve additional objects.</p>"
      }
    },
    "traits": {
      "smithy.api#output": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for this product user.</p>",
            "smithy.api#required": {}
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
            "smithy.api#required": {}
          }
        },
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
            "smithy.api#required": {}
          }
        },
        "Status": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The status of a product for this user.</p>",

```

```

        "smithy.api#required": {}
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {

```

```

"type": "structure",
"members": {
  "RdsSalCredentialsProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ]
  },

```

```

"traits": {
  "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/RegisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummary": {

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
      "smithy.api#error": "client",
      "smithy.api#httpError": 404
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
    "type": "structure",
    "members": {
      "SecretId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",

```



```

        "smithy.api#length": {
            "min": 1
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#SecurityGroup":
{
    "type": "string",
    "traits": {
        "smithy.api#length": {
            "min": 5,
            "max": 200
        },
        "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more servers.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoint for activation servers.</p>",

```

```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings
such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC
endpoints. The security group should permit inbound TCP port 1688 communication from resources in the
VPC.</p>"
},
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
{
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity
provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours
to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services

```

Billing. For more information, see <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html> Viewing your monthly charges in the *Amazon Web Services Billing User Guide*.

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",

"uri": "/user/StartProductSubscription"
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  }
}
```

```

        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        },
    },
    "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
            },
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
            },
        ],
    },

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
          }
        }
      },
      "Domain": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
        "ProductUserSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want

```



```

to tag.</p>",
    "smithy.api#httpLabel": {},
    "smithy.api#required": {}
  }
},
"Tags": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
            "smithy.api#httpQuery": "tagKeys",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/identity-provider/UpdateIdentityProviderSettings"

```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
      }
    },
    "UpdateSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
      "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",

```

```

        "traits": {
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
                "smithy.api#required": {}
            }
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    }
}

```

```

    },
    "traits": {
        "smithy.api#documentation": "<p>A parameter is not valid.</p>",
        "smithy.api#error": "client"
    }
}
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //

```

```

The ID of the workspace to remove the Grafana Enterprise license from.
//
// This member is required.
WorkspaceId *string

noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

// A structure containing information about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
}

```

```

if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {

```



```

    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

}

out := result.(*GetLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err
!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);

```

```

err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:

```

```

python-version: [3.9]

steps:
  - name: Checkout target
    uses: actions/checkout@v2
    with:
      path: sdkbase
      ref: ${ github.base_ref }
  - name: Checkout this ref
    uses: actions/checkout@v2
    with:
      path: new-ref
      fetch-depth: 0
  - name: Get Diff
    run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
  - name: Get Target Files
    run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
  - name: Checkout scancode
    uses: actions/checkout@v2
    with:
      repository: nexB/scancode-toolkit
      path: scancode-toolkit
      fetch-depth: 1
  - name: Set up Python
${{ matrix.python-version }}
    uses: actions/setup-python@v2
    with:
      python-version: ${{ matrix.python-version }}
  # ScanCode
  - name: Self-configure scancode
    working-directory: ./scancode-toolkit
    run: ./scancode --help
  - name: Run Scan code on target
    run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
  - name: Run Scan code on pr ref
    run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
  # compare
  - name: License test
    run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

```



```

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

```

```

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
}

```

```

if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{

```

```

    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
  }
}
AWS SDK for Go
Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Copyright 2014-2015 Stripe, Inc.
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Checks out the specified license for offline use.
```

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CheckoutBorrowLicenseInput struct {
```

```

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

```

```

// Digital signature method. The possible value is JSON Web Signature (JWS)
// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
//
// [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
//
// This member is required.
DigitalSignatureMethod types.DigitalSignatureMethod

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

```

```

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}

```

```

}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
}

```



```

if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }
}

```

```

out := result.(*AssociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
}

```

```

err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

```

```

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
}

```



```

if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {

```

```

return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
return err
}
if err = addInterceptExecution(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
return err
}
if err = addInterceptTransmit(stack, options); err != nil {
return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
return err
}
}

```

```

return nil
}

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}

```

CheckInLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks in the specified license. Check in a license when it is no longer in use.
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
}

```

```

if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}

```


CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/DeregisterSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {

```

```

    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider
resource to deregister.</p>",
          "smithy.api#required": {}
        }
      },
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of name to filter by.</p>"
        }
      },
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe
operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
      "type":
    "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
      },
      "output": {
        "target":
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/GetRegisteredSubscriptionProvider"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {

```



```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      }
    }
  }
}

```

```

    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
      }
    },
    "Region": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
      }
    },
    "UsageOperation": {
      "target": "smithy.api#String",

      "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see<n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage<n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
      }
    },
    "ProductCode": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
      "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage<n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
      }
    }
  }

```



```

    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },
  "OsVersion": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
    }
  },
  "SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
  },
  "SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
  },
  "DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
    }
  },
  "RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager Linux Subscriptions",
      "arnNamespace": "license-manager-linux-subscriptions",
      "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-linux-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
      ],
      "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
      ]
    },
    "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running\n      Amazon EC2 instances.</p>",
    "smithy.api#title": "AWS License Manager Linux Subscriptions",

```

```

"smithy.rules#endpointRuleSet": {
  "version": "1.0",
  "parameters": {
    "Region": {
      "builtIn": "AWS::Region",
      "required": false,
      "documentation": "The AWS region used to dispatch the request.",
      "type": "String"
    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not support dual-stack, dispatching the request MAY return an error.",
      "type": "Boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    },
    {
      "conditions": [
        {

```

```

        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseFIPS"
            },
            true
        ]
    }
},
{
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseDualStack"
                        },
                        true
                    ]
                }
            ]
        }
    ]
},
{
    "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
    "type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
},
{
    "type": "tree"
}
],
}
],

```

```

    "type": "tree"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Region"
              }
            ]
          }
        ],
        "rules": [
          {
            "conditions": [
              {
                "fn": "aws.partition",
                "argv": [
                  {
                    "ref": "Region"
                  }
                ],
                "assign": "PartitionResult"
              }
            ],
            "rules": [
              {
                "conditions": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseFIPS"
                      },
                      true
                    ]
                  }
                ],
                "rules": [
                  {
                    "fn": "booleanEquals",
                    "argv": [
                      {
                        "ref": "UseDualStack"

```

```

        },
        true
    ]
}

],

"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        }
                    ]
                }
            ],
            "supportsFIPS"
        }
    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
},
],
"rules": [
{
    "conditions": [],
}
],
"rules": [
{

```

```

        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "rules": [
            {
                "conditions": [
                    {
                        "fn": "booleanEquals",
                        "argv": [
                            {
                                "fn": "getAttr",
                                "argv": [

```



```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsDualStack"
                    ]
                }
            ]
        }
    ]
}
],
"rules": [
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "error": "DualStack is enabled but this partition does not support DualStack",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            }
        }
    ]
}

```

```

    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  }
}

```

```

    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
      }
    }
  ]
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },

```

```

{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  }
}

```

```

    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
}

    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
}

```



```

    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },

```

```

        {
            "name": "Disabled",
            "value": "Disabled",
            "documentation": "Disabled LinuxSubscriptionsDiscovery"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        }
    ]
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListLinuxSubscriptionInstances"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "Instances"
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n    \t\define the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },

```

```

    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation":
        "NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
      "type": "structure",
      "members": {
        "Instances": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
          "traits": {
            "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
      },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
    "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
    returned results will include data aggregated across your accounts in\n  Organizations.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListLinuxSubscriptions"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "Subscriptions"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
        that match one or\n  more sets of key-value pairs that you specify. For example, you can filter by the name of\n
        <code>Subscription</code> with an optional operator to see subscriptions that match,\n  partially match, or don't
        match a certain subscription's name.</p>\n  <p>The valid names for this filter are:</p>\n  <ul>\n
        <li>\n  <p>\n  <code>Subscription</code>\n  </p>\n  </li>\n  </ul>\n
        <p>The valid Operators for this filter are:</p>\n  <ul>\n  <li>\n  <p>\n  <code>contains</code>\n  </p>\n  </li>\n  <li>\n  <p>\n  <code>equals</code>\n  </p>\n  </li>\n  <li>\n  <p>\n
      
```

```

<code>Notequal</code>\n          </p>\n          </li>\n        </ul>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",

```

```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderSources": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
        }
      },
      "MaxResults": {

```



```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
        "documentation": "Equal operator"
      },
      {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
      },
      {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
      }
    ],
    "smithy.api#length": {
      "min": 1,
      "max": 20
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled OrganizationIntegration"
      },
      {

```

```

"name": "Disabled",
  "value": "Disabled",
  "documentation": "Disabled OrganizationIntegration"
}
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/RegisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {

```



```

subscription provider \n\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
        }
      },
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
        }
      },
      "SubscriptionProviderStatus": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
        }
      },
      "SubscriptionProviderStatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
        }
      }
    },
  },

```

```

    "LastSuccessfulDataRetrievalTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
      "smithy.api#error": "client"
    }
  },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [
          {
            "name": "InProgress",
            "value": "InProgress",

```

```

        "documentation": "InProgress status"
    },
    {
        "name": "Completed",
        "value": "Completed",
        "documentation": "Completed status"
    },
    {
        "name": "Successful",
        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
      "type": "string",
      "traits": {
        "smithy.api#enum": [

```



```

        {
            "name": "RedHat",
            "value": "RedHat",
            "documentation": "RedHat subscription provider namespace"
        }
    ]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
    },

```


"smithy.api#documentation": "<p>The metadata tags to assign to the Amazon Web Services resource.
Tags are \n\t\t\tformatted as key value pairs.</p>",

```
    "smithy.api#required": {}
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
  "type": "map",
  "key":
{
  "target": "smithy.api#String"
},
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
},
"output": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
  "smithy.api#http": {
    "method": "DELETE",
    "uri": "/tags/{resourceArn}"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
  "type": "structure",

  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    },
    "tagKeys": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

        "smithy.api#input":
    {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
        "type": "structure",
        "members": {},
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/UpdateServiceSettings"
            },
            "smithy.api#idempotent": {}
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
        "type": "structure",
        "members": {
            "LinuxSubscriptionsDiscovery": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
                "traits": {
                    "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",

```

```

        "smithy.api#required":
    {}
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>",
            "smithy.api#required": {}
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
        }
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        },
        "LinuxSubscriptionsDiscoverySettings": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
            "traits": {
                "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
            }
        }
    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    }
}

```

```

    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
      "smithy.api#error": "client"
    }
  }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and

limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

ResponseErrorWrapper

RequestIDRetriever

InterceptAfterDeserialization

OperationDeserializer

InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper

RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.71 aws-sdk-go-v2-feature-s3-manager

1.19.15

1.71.1 Available under license :

GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2

```

disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
}
```

```
type CreateLicenseInput struct {
```

```
    // License beneficiary.
```

```
    //
```

```
    // This member is required.
```

```
    Beneficiary *string
```

```
    // Unique, case-sensitive identifier that you provide to ensure the idempotency of  
    // the request.
```

```
    //
```

```
    //
```

```
    // This member is required.
```

```
    ClientToken *string
```

```
    // Configuration for consumption of the license. Choose a provisional
```

```
    // configuration for workloads running with continuous connectivity. Choose a
```

```
    // borrow configuration for workloads with offline usage.
```

```
    //
```

```
    // This member is required.
```

```
    ConsumptionConfiguration *types.ConsumptionConfiguration
```

```
    // License entitlements.
```

```
    //
```

```
    // This member is required.
```

```
    Entitlements []types.Entitlement
```

```
    // Home Region for the license.
```

```
    //
```

```
    // This member is required.
```

```
    HomeRegion *string
```

```
    // License issuer.
```

```
    //
```

```
    // This member is required.
```

```
    Issuer *types.Issuer
```

```
    // License name.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseName *string
```

```
    // Product name.
```

```
    //
```

```
    // This member is required.
```

```
    ProductName *string
```



```

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
}

```

```

err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{}),
middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger

```

OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart

InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization

```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //

```



```

// - Beneficiary
Filters []types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err =
addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}

DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength

```

```

SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))

```

```

(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.

```

```

NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }

```

```

}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
}

```



```

if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {

```

```
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}
```

DisassociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

```

CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        }
      },
      "ActiveDirectorySettings":
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
      }
    },
    "ActiveDirectoryType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
      "traits": {
        "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
      }
    }
  }
}

```

```

    }
  },
  "IsSharedActiveDirectory": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
    "type": "structure",
    "members": {
      "DomainName": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
        }
      },
      "DomainIpv4List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainIpv6List": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
        "traits": {
          "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active Directory.</p>",
          "smithy.api#length": {
            "min": 1,
            "max": 2
          }
        }
      },
      "DomainCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
    }
  },
  "DomainNetworkSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DomainNetworkSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
  },
  "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",
          "value": "SELF_MANAGED",
          "documentation": "SELF_MANAGED type of Active Directory"
        },
        {
          "name": "AWS_MANAGED",
          "value": "AWS_MANAGED",
          "documentation": "AWS_MANAGED type of Active Directory"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {

```

```

"target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The identity provider for the user.</p>",

      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],

```



```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/CreateLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
        "smithy.api#required": {}
      }
    },
    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
        }
      },
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },

```

```

"output": {
  "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/DeleteLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
  "type": "structure",
  "members": {
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
  "type": "structure",
  "members": {
    "LicenseServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code> resource that was deleted.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    }
  ],

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    }
  ],
  {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/identity-provider/DeregisterIdentityProvider"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    }
  },
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
    }
  }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},

"com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

"target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
            }
        },
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
            }
        }
    }
}

```

```

    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
}

```



```

        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
{
        "Attribute": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
            }
        },
        "Operation": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
            }
        },
        "Value": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>Value of the filter.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results
from a describe or list operation. You can use filters can be used to match a set
of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#FilterList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Filter"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProvider": {
    "type": "union",
    "members": {
        "ActiveDirectoryIdentityProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource
contains settings and other details about a specific Active Directory identity provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Settings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
    }

  },
  "FailureMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
    }
  },
  "OwnerAccountId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Describes an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "Products": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
            "smithy.api#required": {}
        }
    },
    "LastStatusCheckDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date of the last status check.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
      "smithy.api#required": {}
    }
  },
  "InstanceUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that

```

contains the user information for the product subscription.</p>"

```
    }
  },
  "AssociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
```

```

    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:[1,7]:|([0-9a-fA-F]{1,4}:[1,6]:[0-9a-fA-F]{1,4})$)"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
    }
  ],
  "traits": {
    "aws.api#service": {
      "sdkId": "License Manager User Subscriptions",
      "arnNamespace": "license-manager-user-subscriptions",
      "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
    },
    "aws.auth#sigv4": {
      "name": "license-manager-user-subscriptions"
    },
    "aws.protocols#restJson1": {},
    "smithy.api#cors": {
      "additionalAllowedHeaders": [
        "*",
        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id",
        "content-length",
        "x-api-key",
        "authorization",

```



```

        "x-amz-date",
        "x-amz-security-token",
        "Access-Control-Allow-Headers",
        "Access-Control-Allow-Methods",
        "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
}

```

```

    },
    "rules": [
      {
        "conditions": [
          {
            "fn": "isSet",
            "argv": [
              {
                "ref": "Endpoint"
              }
            ]
          }
        ]
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
      },
      {
        "conditions": [],

```

```

        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
        },

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
        ],
        "assign": "PartitionResult"
    }
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                ],
            },
        ],
    },

```

true

```
    ]
  },
  {
    "fn": "booleanEquals",
    "argv": [
      {
        "ref": "UseDualStack"
      },
      true
    ]
  }
],
"rules": [
  {
    "conditions": [
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsFIPS"
            ]
          }
        ]
      },
      {
"fn": "booleanEquals",
        "argv": [
          true,
          {
            "fn": "getAttr",
            "argv": [
              {
                "ref": "PartitionResult"
              },
              "supportsDualStack"
            ]
          }
        ]
      }
    ]
  }
]
```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    }
  ],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",

```



```

{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [
            {
              "ref": "PartitionResult"
            },
            "supportsDualStack"
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
        "properties": {},
        "headers": {}
      },
      "type": "endpoint"
    },
    {
      "type": "tree"
    },
    {
      "conditions": [],
      "error": "DualStack is enabled but this partition does not support DualStack",
      "type": "error"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "endpoint": {
    "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": true
                }
            },
            "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
                },
                "params": {
                    "Region": "us-east-1",
                    "UseFIPS": true,
                    "UseDualStack": false
                }
            }
        }
    ]
}

```



```

    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
        }
      },
      "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"

```

```

    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
}

```

```

    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        }
    }

```

```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS":
false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
}

```

```

    },
    {
      "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,

"UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
      "expect": {
        "endpoint": {
          "url": "https://example.com"
        }
      },
      "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
      }
    },
    {
      "documentation": "For custom endpoint with fips enabled and dualstack disabled",
      "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
  },

```

```

        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",
        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        }
    }
}

```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {
  "type": "structure",
  "members": {
    "IdentityProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
      }
    },
    "ServerType": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>"
      }
    },
    "ServerEndpoint": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint",
      "traits": {
        "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
      }
    },
    "LicenseServerEndpointId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointId",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
      }
    },
    "LicenseServerEndpointArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>"

```

```

resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {

```



```

        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
    },
    {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
    },
    {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            }
        ]
    }
}

```

```

        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServer"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of license server.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
    "traits": {
        "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",

```

```

    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
        "smithy.api#required": {}
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results

```

to return from a single request.</p>"

```
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

  "members": {
    "InstanceSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
```

```

"input": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
            "smithy.api#range": {
                "min": 1,
                "max": 100
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
        "type": "structure",
        "members": {
            "LicenseServerEndpoints": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
                "traits": {
                    "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        }
    }
}

```



```

    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {

```

```

        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {
        "Product": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
},

```

```

    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
      "type": "structure",
      "members": {
        "ProductUserSummaries": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse"
      }
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListUserAssociations"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "InstanceUserSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

    "traits": {
      "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output":
  {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for this user.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {

```

```

        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
        }
    },
    "SubscriptionStartDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The start date of a subscription.</p>"
        }
    },
    "SubscriptionEndDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
        "type": "list",
        "member": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
        "type": "structure",
        "members": {
            "RdsSalCredentialsProvider": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
                "traits": {
                    "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a

```


reference to the credentials provider that's used for RDS license server user administration.</p>",

```
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/identity-provider/RegisterIdentityProvider"
```

```

    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  },
  "Settings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
    "traits": {
      "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
    "traits": {
      "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-
\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
      "Endpoint": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ServerType": {
    "type": "string",

```

```

    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
  {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
      "smithy.api#error": "client"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",
    "members": {
      "Subnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
          "smithy.api#length": {
            "min": 1
          },
          "smithy.api#required": {}
        }
      }
    },
    "SecurityGroupId": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
      "traits": {
        "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings

```

such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.

```
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
```

```

"uri": "/user/StartProductSubscription"
  }
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```



```

    ],
    "traits": {
      "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/StopProductSubscription"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "ProductUserArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
          }
        },
        "Domain": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to stop the product subscription.</p>"
          }
        }
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummary": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
        "traits": {
          "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {

```

```

"min": 0,
  "max": 50
},
"smithy.api#sensitive": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
    "smithy.api#http": {
      "method": "PUT",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "Tags": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the
request.</p>",
        "smithy.api#error": "client"
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
        "smithy.api#http": {
          "method": "DELETE",
          "uri": "/tags/{ResourceArn}"
        },
        "smithy.api#idempotent": {}
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest": {
      "type": "structure",
      "members": {
        "ResourceArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want to remove tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}
          }
        }
      },
      "TagKeys": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#TagKeyList",
        "traits": {
          "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
          "smithy.api#httpQuery": "tagKeys",
          "smithy.api#required": {}
        }
      }
    }
  }

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
    "type": "structure",

```

```

"members": {
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
  },
  "Product": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
  }
},
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
      "smithy.api#required": {}
    }
  }
},
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {

```

```

    "smithy.api#output": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
      "AddSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
          "smithy.api#required": {}
        }
      },
      "RemoveSubnets": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
        "traits": {
          "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
          "smithy.api#required": {}
        }
      },
      "SecurityGroupId": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
        "traits": {
          "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ValidationException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A parameter is not valid.</p>",
      "smithy.api#error": "client"
    }
  }
}

```



```

    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

```

```

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {

```

```

    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if

```

```

err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err

```

```

!= nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
}

```



```

if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                  path: sdkbase
                  ref: ${{ github.base_ref }}

```

```

- name: Checkout this ref
  uses: actions/checkout@v2
  with:
    path: new-ref
    fetch-depth: 0
- name: Get Diff
  run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${ github.event.pull_request.base.sha }
${{ github.sha }} > refDiffFiles.txt
- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutLicenseInput struct {

    //
    // Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.

```

```

ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
}

```

```

err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CheckoutLicense",
    }
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks out the specified license for offline use.

```
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns  
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {  
    if params == nil {  
        params = &CheckoutBorrowLicenseInput{}  
    }
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,  
c.addOperationCheckoutBorrowLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckoutBorrowLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

type CheckoutBorrowLicenseInput struct {

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
//

the request.

//

// This member is required.

ClientToken *string

// Digital signature method. The possible value is JSON Web Signature (JWS)

// algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].

//

// [RFC 7518 Digital Signature with RSASSA-PSS]: <https://tools.ietf.org/html/rfc7518#section-3.5>

//

// This member is required.

DigitalSignatureMethod types.DigitalSignatureMethod


```

// License entitlements. Partial checkouts are not supported.
//
// This member is required.
Entitlements []types.EntitlementData

// Amazon Resource Name (ARN) of the license. The license must use the borrow
// consumption configuration.
//
// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more
// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.

```

```

//
// Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
// trials.
//
// This member is required.
LicenseType types.LicenseType

// The ID of the workspace to associate the license with.
//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
}

```

```

if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,

```

```

c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```



```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licenssemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licenssemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"

```

)

```
// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}
```

```
type DeleteLicenseInput struct {
```

```
    // Amazon Resource Name (ARN) of the license.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseArn *string
```

```
    // Current version of the license.
```

```
    //
```

```
    // This member is required.
```

```
    SourceVersion
```

```
    *string
```

```
    noSmithyDocumentSerde
```

```
}
```

```
type DeleteLicenseOutput struct {
```

```
    // Date when the license is deleted.
```

```
    DeletionDate *string
```

```
    // License status.
```

```
    Status types.LicenseDeletionStatus
```

```
    // Metadata pertaining to the operation's result.
```

```
    ResultMetadata middleware.Metadata
```

```
    noSmithyDocumentSerde
```

```

}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err !=
    nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```

```

}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
}
}

```

```
    OperationName: "DeleteLicense",  
  }  
}  
CheckInLicense  
Initialize stack step  
  spanInitializeStart  
  InterceptExecution  
  RegisterServiceMetadata  
  legacyEndpointContextSetter  
  SetLogger  
  OperationInputValidation  
  spanInitializeEnd  
Serialize stack step  
  spanBuildRequestStart  
  setOperationInput  
  ResolveEndpoint  
  InterceptBeforeSerialization  
  OperationSerializer  
  InterceptAfterSerialization  
Build stack step  
  ClientRequestID  
  ComputeContentLength  
  SetCredentialSourceMiddleware  
  UserAgent  
  AddTimeOffsetMiddleware  
  RecursionDetection  
  spanBuildRequestEnd  
Finalize stack step  
  spanRetryLoop  
  InterceptBeforeRetryLoop  
  Retry  
  InterceptAttempt  
  RetryMetricsHeader  
  ResolveAuthScheme  
  GetIdentity  
  ResolveEndpointV2  
  disableHTTPS  
  ComputePayloadHash  
  setLegacyContextSigningOptions  
  InterceptBeforeSigning  
  Signing  
  InterceptAfterSigning  
Deserialize stack step  
  AddRawResponseToMetadata  
  ErrorCloseResponseBody  
  CloseResponseBody  
  ResponseErrorWrapper  
  RequestIDRetriever
```

InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp    "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```

func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))
(*CheckInLicenseOutput, error) {
    if params == nil {
        params = &CheckInLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
c.addOperationCheckInLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckInLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {

```



```

    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}

```

```

}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation

```

spanInitializeEnd
 Serialize stack step
 spanBuildRequestStart
 setOperationInput
 ResolveEndpoint
 InterceptBeforeSerialization
 OperationSerializer
 InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {

```

    "type": "integer"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
    "type": "long"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```



```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      }
    }
  }

```

```

    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include if\n    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
      }
    },
    "Status": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
      }
    },
    "StatusMessage": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
      "traits": {
        "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
      }
    },
    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      }
    },
    "InstanceType": {

```

```

    "target": "smithy.api#String",
    "traits":
{
    "smithy.api#documentation": "<p>The instance type of the resource.</p>"
}
},
"AccountID": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
},
"Status": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
},
"Region": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
},
"UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
        "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\<a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\</a> operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
},
"ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
        "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\</a> operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
},
"LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
}

```

```

    },
    "SubscriptionName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
      }
    },
    "OsVersion": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your
instance.</p>"
      }
    },
    "SubscriptionProviderCreateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp when you registered
the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
      }
    },
    "SubscriptionProviderUpdateTime": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the
registered \n\t\t\tthird-party Linux subscription provider.</p>"
      }
    },
    "DualSubscription": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for
\n\t\t\tthe same software on your instance.</p>"
      }
    },
    "RegisteredWithSubscriptionProvider": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tta
third-party Linux subscription provider that you've registered with License Manager.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux
subscriptions.</p>"
    }
  }

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation":
        "<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
      "type": "service",
      "version": "2018-05-10",
      "operations": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
        }
      ]
    }
  }

```

```

{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {

```

```

        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
        "type": "Boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "String"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [
                    {
                        "ref": "Endpoint"
                    }
                ]
            }
        ]
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ]
    }
]

```

```

    ],
    "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
    "type": "error"
  },
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              },
              true
            ]
          }
        ]
      }
    ]
  }
],
  "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "endpoint": {
    "url": {
      "ref": "Endpoint"
    },
    "properties": {},
    "headers": {}
  },
  "type": "endpoint"
}
],
"type": "tree"

}
],
"type": "tree"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",

```



```

        "argv": [
            {
                "ref": "Region"
            }
        ]
    },
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseFIPS"
                                },
                                true
                            ]
                        },
                        {
                            "fn": "booleanEquals",
                            "argv": [
                                {
                                    "ref": "UseDualStack"
                                },
                                true
                            ]
                        }
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [

```

```

{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        }
      ]
    }
  ],
  "supportsFIPS"
}
]
},
{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}
],
"rules": [
  {
    "conditions": [],

"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }

```

```

    ],
    "type": "tree"
  }
],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
  "type": "error"
}
],
  "type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseFIPS"
        },
        true
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                },
                "supportsFIPS"
              ]
            },
            true
          ]
        }
      ]
    }
  ]
}

```

```

    ]
    }
  ],
  "rules": [
    {
      "conditions": [],
      "rules": [
        {
          "conditions": [],
          "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
          },
          "type": "endpoint"
        }
      ],
      "type": "tree"
    }
  ],
  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS is enabled
but this partition does not support FIPS",
  "type": "error"
}
],
"type": "tree"
},
{
  "conditions": [
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ]
}
],
"rules": [
  {

```

```

"conditions": [
  {
    "fn": "booleanEquals",
    "argv": [
      true,
      {
        "fn": "getAttr",
        "argv": [
          {

            "ref": "PartitionResult"
          },
          "supportsDualStack"
        ]
      }
    ]
  }
],
"rules": [
  {
    "conditions": [],
    "rules": [

      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"type": "tree"
},
{
  "conditions": [],
  "error": "DualStack is enabled but this partition does not support DualStack",
  "type": "error"
}
],
"type": "tree"
},

```

```

        {
            "conditions": [],
            "rules": [
                {
                    "conditions": [],
                    "endpoint": {
                        "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                        "properties": {},
                        "headers": {}
                    },
                    "type": "endpoint"
                }
            ],
            "type": "tree"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
            "expect": {
                "endpoint": {
                    "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
                }
            },
            "params": {
                "Region": "us-east-1",
                "UseFIPS": true,
                "UseDualStack": true
            }
        }
    ],

```

```

{
  "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
    }
  }
}

```

```

    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {

```



```

"documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
    }
  },

```

```

    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
      }
    }
  },
  {
    "params": {
      "Region": "us-isob-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {

```

```

        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        disabled",
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {
            "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
        },
        "params": {

```

```

    "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",

    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "Enabled",
        "value": "Enabled",
        "documentation": "Enabled LinuxSubscriptionsDiscovery"
      },
      {
        "name": "Disabled",
        "value": "Disabled",
        "documentation": "Disabled LinuxSubscriptionsDiscovery"
      }
    ]
  }
}
},

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
  "type": "structure",
  "members": {
    "SourceRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "OrganizationIntegration": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
      "traits": {
        "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists

```

the running Amazon EC2 instances that were discovered with commercial Linux\n subscriptions.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptionInstances"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Instances"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n      <ul>\n        <li>\n          <p>\n            <code>AccountID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>AmiID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>DualSubscription</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>InstanceID</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>InstanceType</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>ProductCode</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Region</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Status</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>UsageOperation</code>\n          </p>\n          </li>\n      </ul>\n      <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n      <ul>\n        <li>\n          <p>\n            <code>contains</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>equals</code>\n          </p>\n          </li>\n        <li>\n          <p>\n            <code>Notequal</code>\n          </p>\n          </li>\n      </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    },
    "traits": {
        "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
        "type": "structure",
        "members": {
            "Instances": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
                }
            }
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
            }
        ]
    }
}

```

```

    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation":
      "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n  organization, the
      returned results will include data aggregated across your accounts in\n  Organizations.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListLinuxSubscriptions"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "Subscriptions"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n  more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n  partially match, or don't
match a certain subscription's name.</p>\n  <p>The valid names for this filter are:</p>\n  <ul>\n
<li>\n    <p>\n      <code>Subscription</code>\n    </p>\n  </li>\n  </ul>\n
<p>The valid Operators for this filter are:</p>\n  <ul>\n    <li>\n      <p>\n        <code>contains</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>equals</code>\n      </p>\n    </li>\n    <li>\n      <p>\n        <code>Notequal</code>\n      </p>\n    </li>\n  </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    }
  }
}

```



```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "Subscriptions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
    "type": "operation",
    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
    },
    "output": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
    }
  },

```

```

"errors": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/subscription/ListRegisteredSubscriptionProviders"
  },
  "smithy.api#idempotent": {},
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "RegisteredSubscriptionProviders"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\t\t\tin the list.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  }
}

```



```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "EQUAL",
          "value": "Equal",
          "documentation": "Equal operator"
        },
        {
          "name": "NOT_EQUAL",
          "value": "NotEqual",
          "documentation": "Not equal operator"
        },
        {
          "name": "CONTAINS",
          "value": "Contains",
          "documentation": "Contains operator"
        }
      ],
      "smithy.api#length": {
        "min": 1,
        "max": 20
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring Your Own License (BYOL) subscription.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/RegisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderSource": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
        "traits": {
          "smithy.api#documentation": "<p>The supported Linux subscription provider to register.</p>",
          "smithy.api#required": {}
        }
      },
      "SecretArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
        "traits": {

```



```

"type": "structure",
"members": {
  "SubscriptionProviderArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
    "traits": {
      "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderSource": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
    "traits": {
      "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
    }
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  }
},

```



```

    "traits": {
      "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-
\\.]{0,63}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",

```

```

        "value": "Successful",
        "documentation": "Successful status"
    },
    {
        "name": "Failed",
        "value": "Failed",
        "documentation": "Failed status"
    }
]
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#length": {
                "min": 1,
                "max": 100
            }
        }
    },
    "traits": {
        "smithy.api#length": {
            "min": 1,
            "max": 100
        },
        "smithy.api#uniqueItems": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
        "target": "smithy.api#String"
    },
    "value": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The name of the subscription.</p>"
            }
        }
    },
    "traits": {}
},

```

```

    "Type": {
      "target": "smithy.api#String",
      "traits": {

        "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

      }
    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {

```

```

    "type":
"list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "ACTIVE",
                "value": "ACTIVE",
                "documentation": "ACTIVE status"
            },
            {
                "name": "INVALID",
                "value": "INVALID",
                "documentation": "INVALID status"
            },
            {
                "name": "PENDING",
                "value": "PENDING",
                "documentation": "PENDING status"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    },
    "traits": {
        "smithy.api#length": {
            "min": 0,
            "max": 50
        },
        "smithy.api#sensitive": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {

```



```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse": {
      "type": "structure",
      "members": {},
      "traits": {
        "smithy.api#output": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#Tags": {
      "type": "map",
      "key": {
        "target": "smithy.api#String"
      },
      "value": {
        "target": "smithy.api#String"
      },
      "traits": {
        "smithy.api#length": {
          "min": 0,
          "max": 50
        },
        "smithy.api#sensitive": {}
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        }
      ]
    }
  }

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web
Services resource.</p>",
    "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": true
        }
      }
    }
  },
  "LinuxSubscriptionsDiscoverySettings": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
    "traits": {
      "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated"
    }
  }
}

```



```

from.",
    "smithy.api#required": {}
  },
  "AllowUpdate": {
    "target": "smithy.api#Boolean",
    "traits": {
      "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux
subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home
Region.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution

RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.72 alpine-baselayout 3.6.8-r1

1.72.1 Available under license :

Note that many files in here are not copyrighted/licensed, so they are actually in the public domain.

Otherwise:

Copyright 1996-2013 Gentoo Foundation

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they

know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification
follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on
the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate

copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt

otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program),

you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License

and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED

TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you
can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.73 aws-sdk-go-v2-service-internal-checksum 1.9.2

1.73.1 Available under license :

```
GetLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
```



```
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.
```

```
package licenssemanager
```

```
import (
```

```

"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Creates a license.
func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CreateLicenseInput struct {

    // License beneficiary.
    //
    // This member is required.
    Beneficiary *string

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    //
    //
    // This member is required.
    ClientToken *string

    // Configuration for consumption of the license. Choose a provisional
    // configuration for workloads running with continuous connectivity. Choose a
    // borrow configuration for workloads with offline usage.
    //
    // This member is required.
    ConsumptionConfiguration *types.ConsumptionConfiguration

    // License entitlements.

```

```

//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.
//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

```

```

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
        middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
}

```

```

if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err
= addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {

```

```

return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
return err
}
if err = addInterceptAttempt(stack, options); err != nil {
return err
}
if err = addInterceptExecution(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
return err
}
if err = addInterceptTransmit(stack, options); err != nil {
return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return
err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}

```

```

}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
    spanInitializeStart
    InterceptExecution
    RegisterServiceMetadata
    legacyEndpointContextSetter
    SetLogger
    OperationInputValidation
    spanInitializeEnd
Serialize stack step
    spanBuildRequestStart
    setOperationInput
    ResolveEndpoint
    InterceptBeforeSerialization
    OperationSerializer
    InterceptAfterSerialization
Build stack step
    ClientRequestID
    ComputeContentLength
    SetCredentialSourceMiddleware
    UserAgent
    AddTimeOffsetMiddleware
    RecursionDetection
    spanBuildRequestEnd
Finalize stack step
    spanRetryLoop
    InterceptBeforeRetryLoop
    Retry
    InterceptAttempt
    RetryMetricsHeader
    ResolveAuthScheme

```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry

InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd

```

Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }
}

```

```

result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListReceivedLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

```

```

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

```

```

type ListReceivedLicensesOutput struct {

    // Received license details.
    Licenses []types.GrantedLicense

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.

```

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
```

```

    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{

```

```
Region:    region,  
ServiceID: ServiceID,  
OperationName: "ListReceivedLicenses",  
}  
}
```

DeleteLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash

setLegacyContextSigningOptions

InterceptBeforeSigning

Signing

InterceptAfterSigning

Deserialize stack step

AddRawResponseToMetadata

ErrorCloseResponseBody

CloseResponseBody

```

ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,
c.addOperationListLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //

```



```

// - Fingerprint
//
// - Status
Filters
[]types.Filter

// Amazon Resource Names (ARNs) of the licenses.
LicenseArns []string

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListLicensesOutput struct {

// License details.
Licenses []types.License

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack,
options, "ListLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}
}

```

```

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {

```

```

    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "ListLicenses",
    }
}

DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID

```

ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException": {
 "type": "structure",
 "members": {
 "message": {
 "target": "smithy.api#String"
 }
 },
 "traits": {
 "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
 "smithy.api#error": "client"
 }
 }
 }
 }

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
      },
      "ActiveDirectorySettings": {
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
          }
        },
        "ActiveDirectoryType": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType",
          "traits": {
            "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
          }
        },
        "IsSharedActiveDirectory": {
          "target": "smithy.api#Boolean",
          "traits": {
            "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed Active Directory. The default value is false.</p>"
          }
        },
        "traits": {
          "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
        },
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
      "type": "structure",
      "members": {
        "DomainName": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
          }
        },
      },
    },
  },

```

```

    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "SELF_MANAGED",

```

```

        "value": "SELF_MANAGED",
        "documentation": "SELF_MANAGED type of Active Directory"
    },
    {
        "name": "AWS_MANAGED",
        "value": "AWS_MANAGED",
        "documentation": "AWS_MANAGED type of Active Directory"
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#Arn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUser": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
        },
        {
            "target":

```



```

"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/user/AssociateUser"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
    "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {

```

```

    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS) license server.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/CreateLicenseServerEndpoint"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    },
    "LicenseServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
      "traits": {
        "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
        "smithy.api#required": {}
      }
    },
    "Tags": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
          }
        },
        "LicenseServerEndpointArn": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  }

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider": {
    "type": "union",
    "members": {
      "SecretsManagerCredentialsProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#SecretsManagerCredentialsProvider",
        "traits": {
          "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ]
  }
}

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
        "LicenseServerEndpointArn": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
                "smithy.api#required": {}
            }
        },
        "ServerType": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServerType",
            "traits": {
                "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
                "smithy.api#required": {}
            }
        }
    }
},
"traits": {
    "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
        "LicenseServerEndpoint": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
            "traits": {
                "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>  
resource that was deleted.</p>"
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      }
    ],
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>An object that specifies
details for the Active Directory identity provider.</p>"
          }
        },
        "Product": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
          }
        },
        "IdentityProviderArn": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity
provider to deregister.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },

    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    },
  },

```



```

"com.amazonaws.licenseanagerusersubscriptions#Directory": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
  "type": "operation",
  "input": {

    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {

    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
        }
      },
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
        }
      },
      "InstanceUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate
from the EC2 instance.</p>"
        }
      },
      "Domain": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
    "type": "structure",
    "members": {

```

```

    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },

  "traits": {
    "smithy.api#output": {}
  }
},

"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
  }
},

"com.amazonaws.licenseanagerusersubscriptions#Filter": {
  "type": "structure",
  "members": {
    {
      "Attribute": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
        }
      },
      "Operation": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
        }
      }
    }
  },

```

```

    "Value": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>Value of the filter.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Refers to an identity provider.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
      "type": "structure",
      "members": {
        "IdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information about an identity provider.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",

      "smithy.api#required": {}
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
      }
    },
    "FailureMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an identity provider.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList": {
      "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
        "smithy.api#required": {}
      }
    },
    "Products": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
        "smithy.api#required": {}
      }
    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    }
  }
}

```

```

    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name
from the identity provider for the user.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "InstanceId": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",

```

```

    "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
    },
    "InstanceUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
        }
    },
    "AssociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
        }
    },
    "DisassociationDate": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList": {
        "type": "list",
        "member": {

```



```

"target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary"
  },
  },
  "com.amazonaws.licensemanagerusersubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
      "smithy.api#error": "server",
      "smithy.api#httpError": 500
    }
  },
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^(?:(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
    }
  },
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv4List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv4"
    }
  },
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{1,7}[0-9a-fA-F]{1,4}:::[0-9a-fA-F]{1,4}([0-9a-fA-F]{1,4}:{1,7}:|[0-9a-fA-F]{1,4}:{1,6}:|[0-9a-fA-F]{1,4}$)"
    }
  },
  },
  "com.amazonaws.licensemanagerusersubscriptions#IPv6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IPv6"
    }
  },
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AssociateUser"
      }
    ],
  },

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResource"
},
{

```

```

    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager User Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
    "builtIn": "AWS::Region",
    "required": false,
    "documentation": "The AWS region used to dispatch the request.",
    "type": "string"

```

```

    },
    "UseDualStack": {
      "builtIn": "AWS::UseDualStack",
      "required": true,
      "default": false,
      "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
      "type": "boolean"
    },
    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
      "type": "boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "string"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ]
},
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ]
}

```

```

        ]
    }
},
"error": "Invalid Configuration: FIPS and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [

{
        "fn": "booleanEquals",
        "argv": [
            {
                "ref": "UseDualStack"
            },
            true
        ]
    }
},
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{
    "conditions": [],
    "endpoint": {
        "url": {
            "ref": "Endpoint"
        },
    },

    "properties": {},
    "headers": {}
},
    "type": "endpoint"
}
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
}
},

```

```

"rules": [
  {
    "conditions": [
      {
        "fn": "aws.partition",
        "argv": [
          {
            "ref": "Region"
          }
        ],
        "assign": "PartitionResult"
      }
    ],
    "rules": [
      {
        "conditions": [
          {
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseFIPS"
              },
              {
                "ref": "UseDualStack"
              }
            ],
            "fn": "booleanEquals",
            "argv": [
              {
                "ref": "UseDualStack"
              }
            ],
            "fn": "booleanEquals",
            "argv": [
              true,
              {
                "fn": "getAttr",
                "argv": [
                  {

```

```

        "ref": "PartitionResult"
      },
      "supportsFIPS"
    ]
  }
]
},

{
  "fn": "booleanEquals",
  "argv": [
    true,
    {
      "fn": "getAttr",
      "argv": [
        {
          "ref": "PartitionResult"
        },
        "supportsDualStack"
      ]
    }
  ]
}

]
}
],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
      "properties": {},
      "headers": {}
    },
    "type": "endpoint"
  }
],

  "type": "tree"
},
{
  "conditions": [],
  "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",
  "type": "error"
}
],
"type": "tree"

```

```

    },
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "fn": "getAttr",
                  "argv": [
                    {
                      "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                  ]
                },
                true
              ]
            }
          ],
          "rules": [
            {
              "conditions": [],
              "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
              },
              "type": "endpoint"
            }
          ],
          "type": "tree"
        }
      ],
    },
  ],

```



```

        {
            "conditions": [],
            "error": "FIPS is enabled but this partition does not support FIPS",
            "type": "error"
        }
    ],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsDualStack"
                            ]
                        }
                    ]
                }
            ],
            "type": "tree"
        },
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
}
]
},
"smithy.rules#endpointTests": {
    "testCases": [
        {
            "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
    "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",

"UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
}

```

```

    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.
1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS":
true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {

```

```

        "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
"endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS":

```

```

false,
    "UseDualStack": true
}
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS":
false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {

```

```

        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",
    "expect": {
        "error": "Invalid Configuration: Missing Region"
    }
}
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {

```



```

        "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
    },
    "HealthStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus",
        "traits": {
            "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
        }
    },
    "Ipv4Address": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
        }
    },
    "Ipv6Address": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint": {
        "type": "structure",
        "members": {
            "IdentityProviderArn": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
                }
            },
            "ServerType": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
                "traits": {
                    "smithy.api#documentation": "<p>The type of license server.</p>"
                }
            },
            "ServerEndpoint": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
        "traits": {
            "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
        }
    },
    "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
        }
    },
    "LicenseServerEndpointId": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
        }
    },
    "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
        }
    },
    "LicenseServerEndpointProvisioningStatus": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
        "traits": {
            "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
        }
    },
    "LicenseServers": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
        "traits": {
            "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
        }
    },
    "CreationTime": {
        "target": "smithy.api#Timestamp",

        "traits": {
            "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
        }
    }
}

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      },
      {
        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
      },
      {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
      },
      {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
      },
      {
        "name": "DELETED",

```

```

"value": "DELETED",
    "documentation": ""
  }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "HEALTHY",
        "value": "HEALTHY",
        "documentation": ""
      },
      {
        "name": "UNHEALTHY",
        "value": "UNHEALTHY",
        "documentation": ""
      },
      {
        "name": "NOT_APPLICABLE",
        "value": "NOT_APPLICABLE",
        "documentation": ""
      }
    ]
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
  "type": "structure",
  "members": {
    "ServerType": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
      "traits": {
        "smithy.api#documentation": "<p>The type of license server.</p>",
        "smithy.api#required": {}
      }
    },
    "ServerSettings": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
      "traits": {

```

```

        "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
        "smithy.api#required": {}
    }

},
"traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",

```

```

        "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "IdentityProviderSummaries"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest": {
    "type": "structure",
    "members": {
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
            }
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummaries": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request"
            }
        }
    }
}

```

```

criteria.</p>",
    "smithy.api#required": {}
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ]
},

```

```

"traits": {
  "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/instance/ListInstances"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "InstanceSummaries"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "Filters": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
    "traits": {
      "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
  "type": "structure",

```



```

"members": {
  "InstanceSummaries": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    }
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/ListLicenseServerEndpoints"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "LicenseServerEndpoints"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    }
  }
}

```

```

    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      }
    ]
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",

    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListProductSubscriptions"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "ProductUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code></p>"
      }
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  }
}

```

```

    },
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummaries": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    }
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  }
},
"traits": {
  "smithy.api#input": {}
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {

```

```

    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
    "type": "structure",
    "members": {
      "InstanceId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "MaxResults": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",

        "traits": {
          "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        }
      },
      "Filters": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
        "traits": {
          "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
        }
      },
      "NextToken": {

```



```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummaries": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
      "smithy.api#required": {}
    }
  },
  "IdentityProvider": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
    "traits": {
      "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
      "smithy.api#required": {}
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of a product for this user.</p>",
      "smithy.api#required": {}
    }
  },
  "ProductUserArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
    }
  },
  "SubscriptionStartDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The start date of a subscription.</p>"
    }
  },
  "SubscriptionEndDate": {
    "target": "smithy.api#String",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The end date of a subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RdsSalSettings": {
    "type": "structure",
    "members": {
        "RdsSalCredentialsProvider": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider",
            "traits": {
                "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
        }
    ]
}

```

```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/RegisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider to register.</p>",
        "smithy.api#required": {}
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
        "smithy.api#required": {}
      }
    }
  }
}

```

```

    },
    "Settings": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
      "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints.</p>"
      }
    },
    "Tags": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceArn": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
      },
    },
    "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      }
    }
  }

```

```

    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains
credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {
      "min": 5,
      "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint": {
  "type": "structure",
  "members": {
    "Endpoint": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
      }
    }
  }
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
      "RdsSalSettings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RDS_SAL",
          "value": "RDS_SAL",
          "documentation": ""
        }
      ]
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licenseanagerusersubscriptions#Settings": {
    "type": "structure",

```

```

"members": {
  "Subnets": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
    "traits": {
      "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
      "smithy.api#length": {
        "min": 1
      },
      "smithy.api#required": {}
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
    "traits": {
      "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
      "smithy.api#required": {}
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    }
  ]
}

```



```

        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },

    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",

"uri": "/user/StartProductSubscription"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
    "type": "structure",
    "members": {
        "Username": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
                "smithy.api#required": {}
            }
        },
        "IdentityProvider": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
            "traits": {
                "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
                "smithy.api#required": {}
            }
        },
        "Product": {
            "target": "smithy.api#String",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
      "smithy.api#required": {}
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionRequest"
  },
  "output": {

```

```

    "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/StopProductSubscription"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {

```

```

        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
        }
    },
    "ProductUserArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    },
    "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscriptionResponse": {
        "type": "structure",
        "members": {
            "ProductUserSummary": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummary",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the start product subscription operation.</p>",
                    "smithy.api#required": {}
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    },
    "com.amazonaws.licensemanagerusersubscriptions#StringList": {
        "type": "list",

```

```

    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ]
  }
}

```

```

    ],
    "traits": {
      "smithy.api#documentation":
"<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation":
"<p>The tags to apply to the specified resource.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
  },

```

```

"value": {
  "target": "smithy.api#String"
},
"traits": {
  "smithy.api#length": {
    "min": 0,
    "max": 50
  },
  "smithy.api#sensitive": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
    "smithy.api#http": {
      "method": "DELETE",
      "uri": "/tags/{ResourceArn}"
    }
  },

```

```

    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
  "type": "structure",
  "members": {
    "ResourceArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "TagKeys": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
    "traits": {
      "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
      "smithy.api#httpQuery": "tagKeys",
      "smithy.api#required": {}
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    }
  ],

```



```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
    "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn":
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
      "smithy.api#documentation": "<p>Updates the registered identity providers product related

```

configuration settings. You can update any combination of settings in a single operation such as the:

- Subnets which you want to add to provision VPC endpoints.
- Subnets which you want to remove the VPC endpoints from.
- Security group ID which permits traffic to the VPC endpoints.

```

    "smithy.api#required": {}
  }

  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
  "type": "structure",
  "members": {
    "IdentityProviderSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
      "traits": {
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
  "type": "structure",
  "members": {
    "AddSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
        "smithy.api#required": {}
      }
    },
    "RemoveSubnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "SecurityGroupId": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",

```

```

        "traits": {
            "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
        }
    },

    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>A parameter is not valid.</p>",
            "smithy.api#error": "client"
        }
    }
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package grafana
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Removes the Grafana Enterprise license from a workspace.

```

func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)

```

```

if err != nil {
    return nil, err
}

out := result.(*DisassociateLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)

```

```

if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}

```

```

if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);
err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDisassociateLicense(region string)

```

```

*awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "DisassociateLicense",
    }
}

```

```

// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"

```

```

awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.
    Version *string

    noSmithyDocumentSerde
}

type
GetLicenseOutput struct {

    // License details.
    License *types.License

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

```



```

func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err
    != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }
}

```

```

if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:
        region,
    ServiceID:  ServiceID,
    OperationName: "GetLicense",

```

```

}
}
name: License Scan

on: [pull_request]

permissions:
  contents: read

jobs:
  licensescan:
    name: License Scan
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9]

    steps:
      - name: Checkout target
        uses: actions/checkout@v2
        with:
          path: sdkbase
          ref: ${{ github.base_ref }}
      - name: Checkout this ref
        uses: actions/checkout@v2
        with:
          path: new-ref
          fetch-depth: 0
      - name: Get Diff
        run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
        ${{ github.sha }} > refDiffFiles.txt
      - name: Get Target Files
        run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt - > targetFiles.txt
      - name: Checkout scancode
        uses: actions/checkout@v2
        with:
          repository: nexB/scancode-toolkit
          path: scancode-toolkit
          fetch-depth: 1
      - name: Set up Python
        ${{ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${{ matrix.python-version }}
      # ScanCode
      - name: Self-configure scancode
        working-directory: ./scancode-toolkit

```

```

run: ./scancode --help
- name: Run Scan code on target
run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
"context"
"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
if params == nil {
params = &CheckoutLicenseInput{}
}

result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
if err != nil {
return nil, err
}

out := result.(*CheckoutLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

//

```

```

Checkout type.
//
// This member is required.
CheckoutType types.CheckoutType

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
// This member is required.
ClientToken *string

// License entitlements.
//
// This member is required.
Entitlements []types.EntitlementData

// Key fingerprint identifying the license.
//
// This member is required.
KeyFingerprint *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// License beneficiary.
Beneficiary *string

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

// Checkout type.
CheckoutType types.CheckoutType

// Allowed license entitlements.
EntitlementsAllowed []types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

```

```

// Amazon Resource Name (ARN) of the checkout license.
LicenseArn
*string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }

```

```

}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
}

```



```

if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}

```

```

}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

// Checks out the specified license for offline use.

```

func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

}

out := result.(*CheckoutBorrowLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    // the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //
    // This member is required.
    LicenseArn *string

    // Information about constraints.
    CheckoutMetadata []types.Metadata

    // Node ID.
    NodeId *string

    noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

    // Information about constraints.
    CheckoutMetadata []types.Metadata

```

```

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
    if err != nil {
        return
    }
    err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package grafana

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more

```

```

// information, see [Upgrade a workspace to Grafana Enterprise].
//
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))
(*AssociateLicenseOutput, error) {
    if params == nil {
        params = &AssociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx,
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*AssociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type AssociateLicenseInput struct {

    // The type of license to associate with the workspace.
    //
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free
    // trials.
    //
    // This member is required.
    LicenseType types.LicenseType

    // The ID of the workspace to associate the license with.
    //
    // This member is required.
    WorkspaceId *string

    // A token from Grafana Labs that ties your Amazon Web Services account with a
    // Grafana Labs account. For more information, see [Link your account with Grafana Labs].
    //
    // [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html#AMG-workspace-register-enterprise
    GrafanaToken *string

    noSmithyDocumentSerde
}

type AssociateLicenseOutput

```



```

struct {

    // A structure containing data about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack,
options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
}

```

```

if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {

```

```

    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
}

```

```

    if err = addSpanBuildRequestEnd(stack); err != nil {
        return err
    }
    return nil
}

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "AssociateLicense",
    }
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Deletes the specified license.
func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.

```

```

//
// This member is required.
LicenseArn *string

// Current version of the license.
//
// This member is required.
SourceVersion
*string

noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

// Date when the license is deleted.
DeletionDate *string

// License status.
Status types.LicenseDeletionStatus

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {

```

```

    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
}

```

```

if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {

```

```

return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
return err
}
if err = addSpanInitializeStart(stack); err != nil {
return err
}
if err = addSpanInitializeEnd(stack); err != nil {
return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
Region:    region,
ServiceID: ServiceID,
OperationName: "DeleteLicense",
}
}

```

```

CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware

```


UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

// Checks in the specified license. Check in a license when it is no longer in use.

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{  
    }  
}
```

```
    result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,  
c.addOperationCheckInLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
    out := result.(*CheckInLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type CheckInLicenseInput struct {
```

```
    // License consumption token.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseConsumptionToken *string
```

```

// License beneficiary.
Beneficiary *string

noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {
return err
}
if
err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {

```

```

return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err = addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if
err = addCredentialSource(stack, options); err != nil {
return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
return err
}
if err = addRecursionDetection(stack); err != nil {
return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
return err
}

```

```

}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {

```

```

    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

```

```

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
      "type": "integer"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
      "type": "long"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
      ]
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    }
  }
}

```

```

    },
    {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
    "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
        "SubscriptionProviderArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
    "type": "structure",
    "members": {
        "Name": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The type of name to filter by.</p>"
            }
        }
    },
    "traits": {}
},

```



```

    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type":
"list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
},

```

```

    "traits": {
      "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetRegisteredSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
            "smithy.api#required": {}
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
      "type": "structure",
      "members": {
        "SubscriptionProviderArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SubscriptionProviderSource": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
          "traits": {
            "smithy.api#documentation": "<p>The subscription provider for the BYOL registration
resource specified in the request.</p>"
          }
        },
        "SecretArn": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
          "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
          }
        }
      }
    }
  }
}

```



```

    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetServiceSettings"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
    "type": "structure",
    "members": {}
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"
        }
      },
      "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
          "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
      },
      "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
          "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
      }
    },
  },

```

```

    "HomeRegions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
    "type": "structure",
    "members": {
      "AmiId": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
        }
      },
      "InstanceID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
        }
      },
      "InstanceType": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The instance type of the resource.</p>"
        }
      },
      "AccountID": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the instance.</p>"
        }
      },
      "Region": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
        }
      }
    }
  }
}

```



```
"traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\tthird-party Linux subscription provider.</p>"
},
"DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for\n\t\tthe same software on your instance.</p>"
    }
},
"RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from\n\t\tthird-party Linux subscription provider that you've registered with License Manager.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
        "smithy.api#error": "server"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
```

```

"operations": [
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
  },
  {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*"
    ],

```



```

        "content-type",
        "x-amz-content-sha256",
        "x-amz-user-agent",
        "x-amzn-platform-id",
        "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager,
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager Linux Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
        "Region": {
            "builtIn": "AWS::Region",
            "required": false,
            "documentation": "The AWS region used to dispatch the request.",
            "type": "String"
        },
        "UseDualStack": {
            "builtIn": "AWS::UseDualStack",
            "required": true,
            "default": false,
            "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
            "type": "Boolean"
        },
        "UseFIPS": {
            "builtIn": "AWS::UseFIPS",
            "required": true,
            "default": false,
            "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
            "type": "Boolean"
        },
        "Endpoint": {
            "builtIn": "SDK::Endpoint",
            "required": false,
            "documentation": "Override the endpoint used to send this request",
            "type": "String"
        }
    }
},
"rules": [

```

```

{
  "conditions": [
    {
      "fn": "isSet",
      "argv": [
        {
          "ref": "Endpoint"
        }
      ]
    }
  ]
},
"rules": [
  {
    "conditions": [
      {
        "fn": "booleanEquals",
        "argv": [
          {
            "ref": "UseFIPS"
          },
          true
        ]
      }
    ]
  }
],
"error": "Invalid Configuration: FIPS and custom endpoint are not supported",
"type": "error"
},
{
  "conditions": [],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseDualStack"
            },
            true
          ]
        }
      ]
    }
  ]
},
"error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
"type": "error"
},
{

```

```

        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "aws.partition",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ],
                    "assign": "PartitionResult"
                }
            ],
            "rules": [
                {
                    "conditions": [

```

```

    {
      "fn": "booleanEquals",
      "argv": [
        {

"ref": "UseFIPS"
        },
        true
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        {
          "ref": "UseDualStack"
        },
        true
      ]
    }
  ],

  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            true,
            {
              "fn": "getAttr",
              "argv": [
                {
                  "ref": "PartitionResult"
                }
              ]
            }
          ]
        }
      ]
    },
    {
      "fn": "booleanEquals",
      "argv": [
        true,
        {
          "fn": "getAttr",
          "argv": [

```

```

        {
            "ref": "PartitionResult"
        },
        "supportsDualStack"
    ]
}
]
}
],
"rules": [
    {
        "conditions": [],

"rules": [

        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],

    "type": "tree"
    }
    ],
    "type": "tree"
    },
    {
        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
    ],
    "type": "tree"
    },
    {

"conditions": [

        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"

```

```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {

"fn": "booleanEquals",
        "argv": [
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsFIPS"
                ]
            },
            true
        ]
    }
},
],
"rules": [
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

"properties": {},

                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{

```

```

        "conditions": [],
        "error": "FIPS is enabled
but this partition does not support FIPS",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",
                        "argv": [
                            {
                                "ref": "PartitionResult"
                            },
                            "supportsDualStack"
                        ]
                    }
                ]
            }
        ]
    }
],
"rules": [
    {
        "conditions": [],
        "rules": [
            {
                "conditions": [],

```

```

        "endpoint": {
            "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"
}

],
"type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
                "properties": {},
                "headers": {}
            },
            "type": "endpoint"
        }
    ],
    "type": "tree"
}
],
"type": "tree"
},
{
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",

```



```

        "type": "error"
      }
    ],
    "type": "tree"
  }
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
      },
      "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
      }
    }
  ]
}

```

```

    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
      }
    }
  },
  {
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.api.amazonwebservices.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
      }
    },
    "params": {
      "Region": "cn-north-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {

```

```

        "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
},
"params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
}
},
{
    "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
}

```

```

    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-gov-east-1",
      "UseFIPS": false,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
      }
    },
    "params": {
      "Region": "us-iso-east-1",
      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"

```

```

    }
  },
  "params": {
    "Region": "us-iso-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
    }
  },
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
    }
  }
},
{
  "params": {
    "Region": "us-isob-east-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
  "expect": {
    "endpoint": {
      "url": "https://example.com"
    }
  },
  "params": {
    "Region": "us-east-1",
    "UseFIPS": false,
    "UseDualStack": false,
    "Endpoint": "https://example.com"
  }
}

```

```

    }
  },
  {
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
      "endpoint": {
        "url": "https://example.com"
      }
    },
    "params": {
      "UseFIPS": false,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
      "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": false,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
      "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": false,
      "UseDualStack": true,
      "Endpoint": "https://example.com"
    }
  },
  {
    "documentation": "Missing region",
    "expect": {
      "error": "Invalid Configuration: Missing Region"
    }
  }
}

```

```

        }
    ],
    "version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "OrganizationIntegration": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
            "traits": {
                "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    }
}

```

```

    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
      },
      "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
      ],
      "traits": {
        "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
        "smithy.api#http": {
          "code": 200,
          "method": "POST",
          "uri": "/subscription/ListLinuxSubscriptionInstances"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
          "inputToken": "NextToken",
          "outputToken": "NextToken",
          "pageSize": "MaxResults",
          "items": "Instances"
        }
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
      "type": "structure",
      "members": {
        "Filters": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
          "traits": {
            "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n    \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n    \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n    \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n    \tto

```



```

streamline results:


- AccountID
- AmiID
- DualSubscription
- InstanceID
- InstanceType
- ProductCode
- Region
- Status
- UsageOperation



For each filter, you can use one of the following Operator values to define the behavior of the filter:



- contains
- equals
- NotEqual

```

```

    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "The maximum items to return in a request."
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "A token to specify where to start paginating. This is the nextToken from a previously truncated response.",
      "smithy.api#length": {
        "min": 1,
        "max": 16384
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "NextToken length limit is half of ddb accepted limit. Increase this limit if parameters in request increases."
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "An array that contains instance objects."
        }
      }
    },
    "NextToken": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\field
isn't empty, there are additional elements that the service hasn't \n\included in this request. Use this token with the
next request to retrieve \n\tadditional
objects.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation":
            "<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n    organization, the
returned results will include data aggregated across your accounts in\n    Organizations.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/subscription/ListLinuxSubscriptions"
        },
        "smithy.api#idempotent": {},
        "smithy.api#paginated": {
            "inputToken": "NextToken",
            "outputToken": "NextToken",
            "pageSize": "MaxResults",
            "items": "Subscriptions"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {

```

```

    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

            "traits": {
                "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n    more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n    partially match, or don't
match a certain subscription's name.</p>\n    <p>The valid names for this filter are:</p>\n    <ul>\n
<li>\n        <p>\n            <code>Subscription</code>\n        </p>\n    </li>\n    </ul>\n
<p>The valid Operators for this filter are:</p>\n    <ul>\n        <li>\n            <p>\n
<code>contains</code>\n        </p>\n        </li>\n        <li>\n            <p>\n
<code>equals</code>\n        </p>\n        </li>\n        <li>\n            <p>\n
<code>NotEqual</code>\n        </p>\n        </li>\n    </ul>"
            }
        },
        "MaxResults": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
            "traits": {
                "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
                "smithy.api#length": {
                    "min": 1,
                    "max": 16384
                }
            }
        },
        "traits": {
            "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
        }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
        "type": "structure",
        "members": {
            "Subscriptions": {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
                "traits": {
                    "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
                }
            }
        }
    }
}

```

```

    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},

```

"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {

```

  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],

```

```

    "traits": {
      "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/ListRegisteredSubscriptionProviders"
      },
      "smithy.api#idempotent": {},
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "RegisteredSubscriptionProviders"
      }
    }
  }

```



```

        "smithy.api#httpLabel": {},
        "smithy.api#required": {}

    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
    "type": "structure",
    "members": {
        "tags": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
            "traits": {
                "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "EQUAL",
                "value": "Equal",
                "documentation": "Equal operator"
            },
            {
                "name": "NOT_EQUAL",
                "value": "NotEqual",
                "documentation": "Not equal operator"
            },
            {
                "name": "CONTAINS",
                "value": "Contains",
                "documentation": "Contains operator"
            }
        ],
        "smithy.api#length": {
            "min": 1,
            "max": 20
        }
    }
}

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "Enabled",
          "value": "Enabled",
          "documentation": "Enabled OrganizationIntegration"
        },
        {
          "name": "Disabled",
          "value": "Disabled",
          "documentation": "Disabled OrganizationIntegration"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Register the supported third-party subscription provider for your Bring

```



```

    }
  },
  "SubscriptionProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
    }
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License
Manager currently supports \n\t\t\t\tRed Hat subscriptions.</p>"
      }
    },
    "SecretArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret
that stores your registered Linux subscription provider \n\t\t\t\taccess token. For RHEL account subscriptions, this is
the offline token.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",

```

```

    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider
access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions,
\n\t\t\tthis is the status of the offline token.</p>"
    }
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription
\n\t\t\tprovider token status.</p>"
    }
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed
third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and
license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider,
License Manager can get subscription data from the registered provider.</p>"
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException": {
    {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {

```

```

    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{{0,63}}:secret:[^/]{1,1023}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "InProgress",
          "value": "InProgress",
          "documentation": "InProgress status"
        },
        {
          "name": "Completed",
          "value": "Completed",
          "documentation": "Completed status"
        },
        {
          "name": "Successful",
          "value": "Successful",
          "documentation": "Successful status"
        },
        {
          "name": "Failed",
          "value": "Failed",
          "documentation": "Failed status"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    }
  },
  "traits": {
    "smithy.api#length": {
      "min": 1,
      "max": 100
    }
  }
}

```

```

    },
    "smithy.api#uniqueItems": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
  "type": "map",
  "key": {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the subscription.</p>"
      }
    },
    "Type": {
      "target": "smithy.api#String",
      "traits": {

```

"smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"

```

    }
  },
  "InstanceCount": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
    "traits": {
      "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
  "type": "list",

  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "RedHat",
          "value": "RedHat",
          "documentation": "RedHat subscription provider namespace"
        }
      ]
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
    "type":
"list",
    "member": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
    "type": "string",
    "traits": {
      "smithy.api#enum": [
        {
          "name": "ACTIVE",
          "value": "ACTIVE",
          "documentation": "ACTIVE status"
        },
        {
          "name": "INVALID",
          "value": "INVALID",
          "documentation": "INVALID status"
        },
        {
          "name": "PENDING",
          "value": "PENDING",
          "documentation": "PENDING status"
        }
      ]
    }
  }
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
  {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Add metadata tags to the specified Amazon Web Services
resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{resourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest": {
    "type": "structure",

```



```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
        },
        {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
        "smithy.api#http": {
            "method": "DELETE",
            "uri": "/tags/{resourceArn}"
        },
        "smithy.api#idempotent": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

"members": {
    "resourceArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
            "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services resource to remove the metadata tags from.</p>",
            "smithy.api#httpLabel": {},
            "smithy.api#required": {}

```



```

        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
                "smithy.api#required":
            {}
        }
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\ integration has been enabled, and which Regions data will be aggregated from.</p>",
            "smithy.api#required": {}
        }
    },
    "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
            "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you allow updates, you can aggregate Linux subscription data in more than one home Region.</p>"
        }
    }
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
    "type": "structure",
    "members": {
        "LinuxSubscriptionsDiscovery": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
            "traits": {
                "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
            }
        }
    },
    "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
        }

    },
    "Status": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
        "traits": {
            "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
        }
    },
    "StatusMessage": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
        "traits": {
            "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
        }
    },
    "HomeRegions": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
        "traits": {
            "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
        }
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID

ComputeContentLength

SetCredentialSourceMiddleware

UserAgent

AddTimeOffsetMiddleware

RecursionDetection

spanBuildRequestEnd

Finalize stack step

spanRetryLoop

InterceptBeforeRetryLoop

Retry

InterceptAttempt

RetryMetricsHeader

ResolveAuthScheme

GetIdentity

ResolveEndpointV2

disableHTTPS

ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme

GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.74 busybox 1.37.0-r14

1.74.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source
of bzip2 version 1.0.4. bzip2 source is distributed
under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2006 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.
jseward@bzip.org
bzip2/libbzip2 version 1.0.4 of 20 December 2006
--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide

a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free

Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where

the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice

like this

when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of
Vice

This General Public License does not permit incorporating your program into

proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.75 aws-sdk-go-v2-service-ssooidc 1.35.3

1.75.1 Available under license :

- GetLicense
- Initialize stack step
 - spanInitializeStart
 - InterceptExecution
 - RegisterServiceMetadata
 - legacyEndpointContextSetter
 - SetLogger
 - OperationInputValidation
 - spanInitializeEnd
- Serialize stack step
 - spanBuildRequestStart
 - setOperationInput
 - ResolveEndpoint
 - InterceptBeforeSerialization
 - OperationSerializer
 - InterceptAfterSerialization
- Build stack step
 - ClientRequestID
 - ComputeContentLength
 - SetCredentialSourceMiddleware
 - UserAgent
 - AddTimeOffsetMiddleware
 - RecursionDetection
 - spanBuildRequestEnd
- Finalize stack step
 - spanRetryLoop
 - InterceptBeforeRetryLoop
 - Retry
 - InterceptAttempt
 - RetryMetricsHeader
 - ResolveAuthScheme
 - GetIdentity
 - ResolveEndpointV2
 - disableHTTPS
 - ComputePayloadHash
 - setLegacyContextSigningOptions
 - InterceptBeforeSigning
 - Signing
 - InterceptAfterSigning

```

Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```
// Creates a license.
```

```

func (c *Client) CreateLicense(ctx context.Context, params *CreateLicenseInput, optFns ...func(*Options))
(*CreateLicenseOutput, error) {
    if params == nil {
        params = &CreateLicenseInput{}
    }

```

```

    result, metadata, err := c.invokeOperation(ctx, "CreateLicense", params, optFns,
c.addOperationCreateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

```

```

    out := result.(*CreateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

```

```
type CreateLicenseInput struct {
```

```

    // License beneficiary.
    //

```

```

// This member is required.
Beneficiary *string

// Unique, case-sensitive identifier that you provide to ensure the idempotency of
// the request.
//
//
This member is required.
ClientToken *string

// Configuration for consumption of the license. Choose a provisional
// configuration for workloads running with continuous connectivity. Choose a
// borrow configuration for workloads with offline usage.
//
// This member is required.
ConsumptionConfiguration *types.ConsumptionConfiguration

// License entitlements.
//
// This member is required.
Entitlements []types.Entitlement

// Home Region for the license.
//
// This member is required.
HomeRegion *string

// License issuer.
//
// This member is required.
Issuer *types.Issuer

// License name.
//
// This member is required.
LicenseName *string

// Product name.
//
// This member is required.
ProductName *string

// Product SKU.
//
// This member is required.
ProductSKU *string

// Date and time range during which the license is valid, in ISO8601-UTC format.

```

```

//
// This member is required.
Validity *types.DatetimeRange

// Information about
the license.
LicenseMetadata []types.Metadata

// Tags to add to the license. For more information about tagging support in
// License Manager, see the [TagResource]operation.
//
// [TagResource]: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_TagResource.html
Tags []types.Tag

noSmithyDocumentSerde
}

type CreateLicenseOutput struct {

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License status.
Status types.LicenseStatus

// License version.
Version *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCreateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCreateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCreateLicense{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CreateLicense"); err != nil {

```

```

return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err =
    = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}

```



```

}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCreateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCreateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return
err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return
err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opCreateLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID: ServiceID,
        OperationName: "CreateLicense",
    }
}

CheckoutBorrowLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint

```

InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListReceivedLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step

spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
CheckoutLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter

SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists received licenses.
func (c *Client) ListReceivedLicenses(ctx context.Context, params *ListReceivedLicensesInput, optFns
...func(*Options)) (*ListReceivedLicensesOutput, error) {
    if params == nil {
        params = &ListReceivedLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListReceivedLicenses", params, optFns,
c.addOperationListReceivedLicensesMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*ListReceivedLicensesOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type ListReceivedLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - ProductSKU
    //
    // - Status
    //
    // - Fingerprint
    //
    // - IssuerName
    //
    // - Beneficiary
    Filters []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

```

```

// Maximum number of results to return in a single call.
MaxResults *int32

// Token for the next set of results.
NextToken *string

noSmithyDocumentSerde
}

type ListReceivedLicensesOutput struct {

// Received license details.
Licenses []types.GrantedLicense

// Token for the next set of results.
NextToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationListReceivedLicensesMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpListReceivedLicenses{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListReceivedLicenses{},
middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "ListReceivedLicenses"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
return err
}
if err = addClientRequestID(stack); err != nil {

```

```

return err
}
if err = addComputeContentLength(stack); err != nil {
return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
return err
}
if err = addComputePayloadSHA256(stack); err != nil {
return err
}
if err = addRetry(stack, options); err != nil {
return err
}
if err = addRawResponseToMetadata(stack); err != nil {
return err
}
if err = addRecordResponseTiming(stack); err != nil {
return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
return err
}
if err =
addClientUserAgent(stack, options); err != nil {
return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
return err
}
if err = addCredentialSource(stack, options); err != nil {
return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListReceivedLicenses(options.Region),
middleware.Before); err != nil {
return err
}

```



```

}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return
err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {

```

```

    return err
}
if
err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListReceivedLicenses(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListReceivedLicenses",
}
}
DeleteLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step

```

```

spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Lists the licenses for your account.
func (c *Client) ListLicenses(ctx context.Context, params *ListLicensesInput, optFns ...func(*Options))
(*ListLicensesOutput, error) {
    if params == nil {
        params = &ListLicensesInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "ListLicenses", params, optFns,

```

```

c.addOperationListLicensesMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*ListLicensesOutput)
out.ResultMetadata = metadata
return out, nil
}

type ListLicensesInput struct {

    // Filters to scope the results. The following filters are supported:
    //
    // - Beneficiary
    //
    // - ProductSKU
    //
    // - Fingerprint
    //
    // - Status
    Filters
    []types.Filter

    // Amazon Resource Names (ARNs) of the licenses.
    LicenseArns []string

    // Maximum number of results to return in a single call.
    MaxResults *int32

    // Token for the next set of results.
    NextToken *string

    noSmithyDocumentSerde
}

type ListLicensesOutput struct {

    // License details.
    Licenses []types.License

    // Token for the next set of results.
    NextToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde

```

```

}

func (c *Client) addOperationListLicensesMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpListLicenses{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack,
        options, "ListLicenses"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
    if err = addRecordResponseTiming(stack); err != nil {
        return err
    }
    if err = addSpanRetryLoop(stack, options); err != nil {
        return err
    }

```

```

}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opListLicenses(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack,
options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}

```

```

}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opListLicenses(region string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "ListLicenses",
}
}

```

DisassociateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization


```

AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
{
  "smithy": "2.0",
  "shapes": {
    "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException": {
      "type": "structure",
      "members": {
        "message": {
          "target": "smithy.api#String"
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>You don't have sufficient access to perform this action.</p>",
        "smithy.api#error": "client"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryIdentityProvider": {
      "type": "structure",
      "members": {
        "DirectoryId": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Directory",
          "traits": {
            "smithy.api#documentation": "<p>The directory ID for an Active Directory identity provider.</p>"
          }
        },
        "ActiveDirectorySettings": {
          {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectorySettings",
            "traits": {
              "smithy.api#documentation": "<p>The <code>ActiveDirectorySettings</code> resource contains details about the Active Directory, including network access details such as domain name and IP addresses, and the credential provider for user administration.</p>"
            }
          },
          "ActiveDirectoryType": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#ActiveDirectoryType",
            "traits": {
              "smithy.api#documentation": "<p>The type of Active Directory either a self-managed Active Directory or an Amazon Web Services Managed Active Directory.</p>"
            }
          },
          "IsSharedActiveDirectory": {
            "target": "smithy.api#Boolean",
            "traits": {

```

```

    "smithy.api#documentation": "<p>Whether this directory is shared from an Amazon Web Services Managed
Active Directory. The default value is false.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>Details about an Active Directory identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ActiveDirectorySettings": {
  "type": "structure",
  "members": {
    "DomainName": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name for the Active Directory.</p>"
      }
    },
    "DomainIpv4List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV4List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv4
addresses that are used for the Active Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainIpv6List": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6List",
      "traits": {
        "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the Active
Directory.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 2
        }
      }
    },
    "DomainCredentialsProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Points to the <code>CredentialsProvider</code>
resource that contains information about the credential provider for user administration.</p>"
      }
    },
    "DomainNetworkSettings": {

```

```

        "target": "com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings",
        "traits": {
            "smithy.api#documentation": "<p>The <code>DomainNetworkSettings</code> resource contains an
array of subnets that apply for the Active Directory.</p>"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains network access and credential details that are needed for user
administration in the Active Directory.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryType": {
        "type": "string",
        "traits": {
            "smithy.api#enum": [
                {
                    "name": "SELF_MANAGED",
                    "value": "SELF_MANAGED",
                    "documentation": "SELF_MANAGED type of Active Directory"
                },
                {
                    "name": "AWS_MANAGED",
                    "value": "AWS_MANAGED",
                    "documentation": "AWS_MANAGED type of Active Directory"
                }
            ]
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Arn": {
        "type": "string",
        "traits": {
            "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-
zA-Z0-9-\\.]{1,510}/[a-zA-Z0-9-\\.]{1,510}$"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#AssociateUser": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest"
        },
        "output": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUserResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
            }
        ]
    }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target":
"com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Associates the user to an EC2 instance to utilize user-based
subscriptions.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48
hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web
Services Billing. For more information, see <a
href=\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\">Viewing your monthly
charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/AssociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#AssociateUserRequest":
{
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceId": {

```

```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The ID of the EC2 instance that provides the user-based
subscription.</p>",
            "smithy.api#required": {}
        }
    },
    "IdentityProvider": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>The identity provider for the user.</p>",

            "smithy.api#required": {}
        }
    },
    "Domain": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to associate.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply for the user association.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#AssociateUserResponse": {
    "type": "structure",
    "members": {
        "InstanceUserSummary": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},

```

```

"com.amazonaws.licensemanagerusersubscriptions#BoxInteger": {
  "type": "integer"
},
"com.amazonaws.licensemanagerusersubscriptions#ConflictException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request couldn't be completed because it conflicted with the current
state of the resource.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ]
}

```

```

],
"traits": {
  "smithy.api#documentation": "<p>Creates a network endpoint for the Remote Desktop Services (RDS)
license server.</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/CreateLicenseServerEndpoint"
  },
  "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointRequest": {
  "type": "structure",
  "members":
{
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the
<code>IdentityProvider</code> resource that contains details about a registered identity provider. In the case of
Active Directory, that can be a self-managed Active Directory or an Amazon Web Services Managed Active
Directory that contains user identity details.</p>",
      "smithy.api#required": {}
    }
  },
  "LicenseServerSettings": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings",
    "traits": {
      "smithy.api#documentation": "<p>The <code>LicenseServerSettings</code> resource to create for
the endpoint. The settings include the type of license server and the Secrets Manager
secret that enables administrators to add or remove users associated with the license server.</p>",
      "smithy.api#required": {}
    }
  },
  "Tags": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
    "traits": {
      "smithy.api#documentation": "<p>The tags that apply for the license server endpoint.</p>"
    }
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#CreateLicenseServerEndpointResponse": {
  "type": "structure",

```

```

"members": {
  "IdentityProviderArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity
provider specified in the request.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>LicenseServerEndpoint</code>
resource.</p>"
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#CredentialsProvider": {
  "type": "union",
  "members": {
    "SecretsManagerCredentialsProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider",
      "traits": {
        "smithy.api#documentation": "<p>Identifies the Secrets Manager secret that contains credentials
needed for user administration in
the Active Directory.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains information about the credential provider for user
administration.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpoint": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {

"target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deletes a <code>LicenseServerEndpoint</code> resource.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/license-server/DeleteLicenseServerEndpoint"
    },
    "smithy.api#idempotent": {}
  },
  "com.amazonaws.licensemanagerusersubscriptions#DeleteLicenseServerEndpointRequest": {
    "type": "structure",
    "members": {
      "LicenseServerEndpointArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the  
<code>LicenseServerEndpoint</code> resource to delete.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of License Server that the delete request refers to.</p>",
          "smithy.api#required": {}
        }
      }
    }
  }
}

```

```

    }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpointResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoint": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint",
        "traits": {
          "smithy.api#documentation": "<p>Shows details from the <code>LicenseServerEndpoint</code>
resource that was deleted.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      }
    ]
  }
}

```

```

    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Deregisters the Active Directory identity provider from License Manager user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/DeregisterIdentityProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#DeregisterIdentityProviderRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "IdentityProviderArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the identity provider to deregister.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
}

```

```

    },

    "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProviderResponse": {
      "type": "structure",
      "members": {
        "IdentityProviderSummary": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
          "traits": {
            "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider operation.</p>",
            "smithy.api#required": {}
          }
        },
      },
      "traits": {
        "smithy.api#output": {}
      }
    },

    "com.amazonaws.licenseanagerusersubscriptions#Directory": {
      "type": "string",
      "traits": {
        "smithy.api#pattern": "^(d|sd)-[0-9a-f]{10}$"
      }
    },

    "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser": {
      "type": "operation",
      "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest"
      },
      "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse"
      },
      "errors": [
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        }
      ]
    }
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Disassociates the user from an EC2 instance providing user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/DisassociateUser"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the Active Directory
identity provider for the user.</p>"
      }
    },
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance which provides user-based
subscriptions.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the Active Directory identity
provider.</p>"
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the user to disassociate

```

from the EC2 instance.</p>"

```
    }
  },
  "Domain": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The domain name of the Active Directory that contains
information for the user to disassociate.</p>"
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DisassociateUserResponse": {
  "type": "structure",
  "members": {
    "InstanceUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the associate user operation.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#DomainNetworkSettings": {
  "type": "structure",
  "members": {
    "Subnets": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
      "traits": {
        "smithy.api#documentation": "<p>Contains a list of subnets that apply for the Active Directory
domain.</p>",
        "smithy.api#length": {
          "min": 1
        },
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains network settings for the Active Directory domain.</p>"
  }
}
```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Filter": {
    "type": "structure",
    "members": {
      {
        "Attribute": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The name of an attribute to use as a filter.</p>"
          }
        },
        "Operation": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The type of search (For example, eq, geq, leq)</p>"
          }
        },
        "Value": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>Value of the filter.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#documentation": "<p>A filter name and value pair that is used to return more specific results from a describe or list operation. You can use filters can be used to match a set of resources by specific criteria, such as tags, attributes, or IDs.</p>"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#FilterList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Filter"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider": {
      "type": "union",
      "members": {
        "ActiveDirectoryIdentityProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ActiveDirectoryIdentityProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>ActiveDirectoryIdentityProvider</code> resource contains settings and other details about a specific Active Directory identity provider.</p>"
          }
        }
      }
    },
  },

```

```

    "traits": {
      "smithy.api#documentation": "<p>Refers to
an identity provider.</p>"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary": {
    "type": "structure",
    "members": {
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource contains information
about an identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Settings": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Settings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>Settings</code> resource contains details about the
registered identity providers product related configuration settings, such as the subnets to provision VPC
endpoints.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProviderArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider.</p>"
        }
      }
    },
    "FailureMessage": {

```



```

        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The failure message associated with an identity provider.</p>"
        }
    },
    "OwnerAccountId": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
        }
    }
},
"traits": {
    "smithy.api#documentation": "<p>Describes an identity provider.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceSummary":
{
    "type": "structure",
    "members": {
        "InstanceId": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
                "smithy.api#required": {}
            }
        },
        "Status": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The status of an EC2 instance resource.</p>",
                "smithy.api#required": {}
            }
        },
        "Products": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>A list of provided user-based subscription
products.</p>",
                "smithy.api#required": {}
            }
        }
    }
}

```

```

    },
    "LastStatusCheckDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date of the last status check.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for an EC2 instance.</p>"
      }
    },
    "OwnerAccountId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AWS Account ID of the owner of this resource.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details about the identity provider.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Describes an EC2 instance providing user-based subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceSummary"
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummary": {
      "type": "structure",
      "members": {
        "Username": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
  },

```

```

    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance that provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>The <code>IdentityProvider</code> resource specifies details
about the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "Status": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status of a user associated with an EC2 instance.</p>",
        "smithy.api#required": {}
      }
    },
    "InstanceUserArn": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) that identifies the instance
user.</p>"
      }
    },
    "StatusMessage": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The status message for users of an EC2 instance.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that
contains the user information for the product subscription.</p>"
      }
    },
    "AssociationDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The date a user was associated with an EC2 instance.</p>"
      }
    }
  }

```

```

    }
  },
  "DisassociationDate": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The date a user was disassociated from an EC2 instance.</p>"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Describes users of an EC2 instance providing user-based
subscriptions.</p>"
}
},
"com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummaryList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceUserSummary"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>An exception occurred with the service.</p>",
  "smithy.api#error": "server",
  "smithy.api#httpError": 500
}
},
"com.amazonaws.licenseanagerusersubscriptions#IPv4": {
  "type": "string",
  "traits": {
    "smithy.api#pattern": "^(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])(\\.(?!$)|$)){4}$"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv4List": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#IPv4"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#IPv6": {
  "type": "string",

```

```

    "traits": {
      "smithy.api#pattern": "^[0-9a-fA-F]{1,4}:{7}[0-9a-fA-F]{1,4}|:::[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:{1,7}):|([0-9a-fA-F]{1,4}):{1,6}:|0-9a-fA-F]{1,4}$"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#IpV6List": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IpV6"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#LicenseManagerUserSubscriptions": {
    "type": "service",
    "version": "2018-05-10",
    "operations": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AssociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#CreateLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeleteLicenseServerEndpoint"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DeregisterIdentityProvider"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#DisassociateUser"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProviders"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstances"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListTagsForResource"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ListUserAssociations"
      }
    ]
  }
}

```

```

{
  "target": "com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProvider"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StartProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#StopProductSubscription"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#TagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResource"
},
{
  "target": "com.amazonaws.licensemanagerusersubscriptions#UpdateIdentityProviderSettings"
}
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager User Subscriptions",
    "arnNamespace": "license-manager-user-subscriptions",
    "cloudTrailEventSource": "license-manager-user-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-user-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id",
      "content-length",
      "x-api-key",
      "authorization",
      "x-amz-date",
      "x-amz-security-token",
      "Access-Control-Allow-Headers",
      "Access-Control-Allow-Methods",
      "Access-Control-Allow-Origin"
    ],
    "additionalExposedHeaders": [

```

```

        "x-amzn-errortype",
        "x-amzn-requestid",
        "x-amzn-trace-id"
    ]
},
"smithy.api#documentation": "<p>With License Manager, you can create user-based subscriptions to
utilize licensed software with a per user subscription fee on Amazon EC2 instances.</p>",
"smithy.api#title": "AWS License Manager User Subscriptions",
"smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {

"Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "string"
    },
    "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not
support dual-stack, dispatching the request MAY return an error.",
        "type": "boolean"
    },
    "UseFIPS": {
        "builtIn": "AWS::UseFIPS",
        "required": true,
        "default": false,
        "documentation": "When true, send this request to
the FIPS-compliant regional endpoint. If the configured endpoint does not have a FIPS compliant endpoint,
dispatching the request will return an error.",
        "type": "boolean"
    },
    "Endpoint": {
        "builtIn": "SDK::Endpoint",
        "required": false,
        "documentation": "Override the endpoint used to send this request",
        "type": "string"
    }
},
"rules": [
    {
        "conditions": [
            {
                "fn": "isSet",
                "argv": [

```

```

        {
            "ref": "Endpoint"
        }
    ]
}

],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseFIPS"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        }
    }
]
}

```



```

        },
        "type": "endpoint"
    }
},
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "isSet",
            "argv": [
                {
                    "ref": "Region"
                }
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "aws.partition",
                "argv": [
                    {
                        "ref": "Region"
                    }
                ]
            },
            "assign": "PartitionResult"
        ]
    },
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "ref": "UseFIPS"
                        },
                    ],
                }
            ],
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                }
            ]
        }
    ]
},
{
    "fn": "booleanEquals",
    "argv": [
        {
            "ref": "UseDualStack"
        }
    ]
}

```

```

        },
        true
    ]
}
],
"rules": [
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                true,
                {
                    "fn": "getAttr",
                    "argv": [
                        {
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                }
            ]
        }
    ],
    },
    {
        "fn": "booleanEquals",
        "argv": [
            true,
            {
                "fn": "getAttr",
                "argv": [
                    {
                        "ref": "PartitionResult"
                    },
                    "supportsDualStack"
                ]
            }
        ]
    }
    ],
    },
    "rules": [
        {
            "conditions": [],
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",

```

```

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}

],

    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS and DualStack are enabled, but this partition does not support one or
both",

    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [

{
            "ref": "UseFIPS"
        },
        true
    ]
}
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "fn": "getAttr",
                        "argv": [

{
                            "ref": "PartitionResult"
                        },
                        "supportsFIPS"
                    ]
                },
                true
            ]
        }
    ]
}

```

```

        ]
    }
},
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.{Region}.{PartitionResult#dnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
},
"type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled but this partition does not support FIPS",
    "type": "error"
}

],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
},
],
"rules": [
    {
        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    true,
                    {
                        "fn": "getAttr",

```

```

        "argv": [
            {
                "ref": "PartitionResult"
            },
            "supportsDualStack"
        ]
    }

    ]
}
],
"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }

    ],
    "type": "tree"
},
{
    "conditions": [],
    "error": "DualStack is enabled but this partition does not support DualStack",
    "type": "error"
}
],
"type": "tree"
},
{
    "conditions": [],
    "endpoint": {
        "url": "https://license-manager-user-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},
        "headers": {}
    },
    "type": "endpoint"
}
],
"type": "tree"
}

```

```

    ],
    "type": "tree"
  },
  {
    "conditions": [],
    "error": "Invalid Configuration: Missing Region",
    "type": "error"
  }
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",

"expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.api.aws"
      }
    },
    "params": {
      "Region": "us-east-1",
      "UseFIPS": true,
      "UseDualStack": true
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions-fips.us-east-1.amazonaws.com"
      }
    },
    "params": {
      "Region": "us-east-1",

      "UseFIPS": true,
      "UseDualStack": false
    }
  },
  {
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
      "endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.api.aws"
      }
    },
    "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {

"endpoint": {
        "url": "https://license-manager-user-subscriptions.us-east-1.amazonaws.com"
    }
},
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS":
true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-user-subscriptions-fips.cn-north-1.amazonaws.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{

```

```

        "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS": false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.cn-north-1.amazonaws.com.cn"
            }
        },
        "params": {
            "Region": "cn-north-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-gov-east-1.amazonaws.com"
            }
        },
    },

```



```

        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.api.aws"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS":
false,
            "UseDualStack": true
        }
    },
    {
        "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-gov-east-1.amazonaws.com"
            }
        },
        "params": {
            "Region": "us-gov-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint":
{
                "url": "https://license-manager-user-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {

```

```

        "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-iso-east-1.c2s.ic.gov"
            }
        },
        "params": {
            "Region": "us-iso-east-1",
            "UseFIPS":
false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": true,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
        "expect": {
            "endpoint": {
                "url": "https://license-manager-user-subscriptions.us-isob-east-1.sc2s.sgov.gov"
            }
        },
        "params": {
            "Region": "us-isob-east-1",
            "UseFIPS": false,
            "UseDualStack": false
        }
    },
    {
        "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {

```

```

        "Region": "us-east-1",
        "UseFIPS": false,

        "UseDualStack": false,
        "Endpoint": "https://example.com"
    },
    {
        "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
        "expect": {
            "endpoint": {
                "url": "https://example.com"
            }
        },
        "params": {
            "UseFIPS": false,
            "UseDualStack": false,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "For custom endpoint with fips enabled and dualstack disabled",
        "expect": {

"error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",
    "expect": {
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "Missing region",

```

```

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    },
    "version": "1.0"
}

},
"com.amazonaws.licenseanagerusersubscriptions#LicenseServer": {
    "type": "structure",
    "members": {
        "ProvisioningStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
            "traits": {
                "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server.</p>"
            }
        },
        "HealthStatus": {
            "target":
"com.amazonaws.licenseanagerusersubscriptions#LicenseServerHealthStatus",
            "traits": {
                "smithy.api#documentation": "<p>The health status of the RDS license server.</p>"
            }
        },
        "Ipv4Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv4 addresses that are used for the RDS license
server.</p>"
            }
        },
        "Ipv6Address": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A list of domain IPv6 addresses that are used for the RDS license
server.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Information about a Remote Desktop Services (RDS) license
server.</p>"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpoint": {

```

```

"type": "structure",
"members": {
  "IdentityProviderArn": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider that's
associated with the RDS license server endpoint.</p>"
    }
  },
  "ServerType": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
    "traits": {
      "smithy.api#documentation": "<p>The type of license server.</p>"
    }
  },
  "ServerEndpoint": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ServerEndpoint",
    "traits": {
      "smithy.api#documentation": "<p>The
<code>ServerEndpoint</code> resource contains the network address of the RDS license server endpoint.</p>"
    }
  },
  "StatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The message associated with the provisioning status, if there is
one.</p>"
    }
  },
  "LicenseServerEndpointId": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId",
    "traits": {
      "smithy.api#documentation": "<p>The ID of the license server endpoint.</p>"
    }
  },
  "LicenseServerEndpointArn": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#Arn",
    "traits": {
      "smithy.api#documentation": "<p>The ARN of the <code>ServerEndpoint</code>
resource for the RDS license server.</p>"
    }
  },
  "LicenseServerEndpointProvisioningStatus": {
    "target":
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus",
    "traits": {
      "smithy.api#documentation": "<p>The current state of the provisioning process for the RDS license
server endpoint</p>"
    }
  }
}

```

```

    }
  },
  "LicenseServers": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerList",
    "traits": {
      "smithy.api#documentation": "<p>An array of <code>LicenseServer</code> resources that represent
the license servers that are accessed through this endpoint.</p>"
    }
  },
  "CreationTime": {
    "target": "smithy.api#Timestamp",

    "traits": {
      "smithy.api#documentation": "<p>The timestamp when License Manager created the license server
endpoint.</p>"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>Contains details about a network endpoint for a Remote Desktop
Services (RDS) license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointId": {
  "type": "string"
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpoint"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerEndpointProvisioningStatus": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "PROVISIONING",
        "value": "PROVISIONING",
        "documentation": ""
      },
      {
        "name": "PROVISIONING_FAILED",
        "value": "PROVISIONING_FAILED",
        "documentation": ""
      }
    ]
  }
}

```

```

        "name": "PROVISIONED",
        "value": "PROVISIONED",
        "documentation": ""
    },
    {
        "name": "DELETING",
        "value": "DELETING",
        "documentation": ""
    },
    {
        "name": "DELETION_FAILED",
        "value": "DELETION_FAILED",
        "documentation": ""
    },
    {
        "name": "DELETED",
        "value": "DELETED",
        "documentation": ""
    }
]
}
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerHealthStatus": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "HEALTHY",
                "value": "HEALTHY",
                "documentation": ""
            },
            {
                "name": "UNHEALTHY",
                "value": "UNHEALTHY",
                "documentation": ""
            },
            {
                "name": "NOT_APPLICABLE",
                "value": "NOT_APPLICABLE",
                "documentation": ""
            }
        ]
    }
},
"com.amazonaws.licensemanagerusersubscriptions#LicenseServerList": {
    "type": "list",

```

```

    "member": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#LicenseServer"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#LicenseServerSettings": {
    "type": "structure",
    "members": {
      "ServerType": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerType",
        "traits": {
          "smithy.api#documentation": "<p>The type of license server.</p>",
          "smithy.api#required": {}
        }
      },
      "ServerSettings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ServerSettings",
        "traits": {
          "smithy.api#documentation": "<p>The <code>ServerSettings</code> resource contains the settings
for your server.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The settings to configure your license server.</p>"
  }
},
"com.amazonaws.licensemanagerusersubscriptions#ListIdentityProviders": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerusersubscriptions#ListIdentityProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the Active Directory identity providers for user-based
subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/identity-provider/ListIdentityProviders"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "IdentityProviderSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Product</p> </li> <li> <p>DirectoryId</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken

```

from a previously truncated response.</p>"

```
    }
  },
  "traits": {
    "smithy.api#input": {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListIdentityProvidersResponse": {
    "type": "structure",
    "members": {
      "IdentityProviderSummaries": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummaryList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>IdentityProviderSummary</code> resources
that contain details about the Active Directory identity providers that meet the request
criteria.</p>",
          "smithy.api#required": {}
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListInstances": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      }
    ]
  }
}
```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists the EC2 instances providing user-based subscriptions.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/instance/ListInstances"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListInstancesRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results
to return from a single request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>InstanceId</p> </li> </ul>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    },
    "com.amazonaws.licenseanagerusersubscriptions#ListInstancesResponse": {
      "type": "structure",

      "members": {
        "InstanceSummaries": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#InstanceSummaryList",
          "traits": {
            "smithy.api#documentation": "<p>An array of <code>InstanceSummary</code> resources that
contain details about the instances that provide user-based subscriptions and also meet the request criteria.</p>"
          }
        },
        "NextToken": {
          "target": "smithy.api#String",
          "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      },
      "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpoints": {
        "type": "operation",
        "input": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest"
        },
        "output": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse"
        },
        "errors": [
          {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
          }
        ]
      }
    }
  }
}

```

```

    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
  ],
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>List the Remote Desktop Services (RDS) License Server endpoints
</p>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
    "uri": "/license-server/ListLicenseServerEndpoints"
  },
  "smithy.api#paginated": {
    "inputToken": "NextToken",
    "outputToken": "NextToken",
    "pageSize": "MaxResults",
    "items": "LicenseServerEndpoints"
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsRequest": {
  "type": "structure",
  "members": {
    "MaxResults": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>",
        "smithy.api#range": {
          "min": 1,
          "max": 100
        }
      }
    }
  },

```

```

    "Filters": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>IdentityProviderArn</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken from
a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListLicenseServerEndpointsResponse": {
    "type": "structure",
    "members": {
      "LicenseServerEndpoints": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#LicenseServerEndpointList",
        "traits": {
          "smithy.api#documentation": "<p>An array of <code>LicenseServerEndpoint</code> resources that
contain detailed information about the RDS License Servers that meet the request criteria.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest"
    }
  }
}

```

```

    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the user-based subscription products available from an identity provider.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/user/ListProductSubscriptions"
      },
      "smithy.api#paginated": {
        "inputToken": "NextToken",
        "outputToken": "NextToken",
        "pageSize": "MaxResults",
        "items": "ProductUserSummaries"
      }
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#ListProductSubscriptionsRequest": {
    "type": "structure",
    "members": {

```

```

    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code>
</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
      }
    },
    "Filters": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
      "traits": {
        "smithy.api#documentation": "<p>You can use the following filters to streamline
results:</p> <ul> <li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListProductSubscriptionsResponse": {
    "type": "structure",
    "members": {
      "ProductUserSummaries": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ProductUserSummaryList",
        "traits": {

```



```

        "smithy.api#documentation": "<p>Metadata that describes the list product subscriptions
operation.</p>"
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
        }
    },
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResource": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse"
    }
},
"errors": [
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
    },
    {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ValidationException"
    }
],
"traits": {
    "smithy.api#documentation": "<p>Returns the list of tags for the specified resource.</p>",
    "smithy.api#http": {
        "method": "GET",
        "uri": "/tags/{ResourceArn}"
    },
    "smithy.api#readonly": {}
}
},
"com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceRequest": {
    "type": "structure",

```

```

    "members": {
      "ResourceArn": {
        "target":
"com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource whose tags
you want to retrieve.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "traits": {
        "smithy.api#input": {}
      }
    },
    "com.amazonaws.licensemanagerusersubscriptions#ListTagsForResourceResponse": {
      "type": "structure",
      "members": {
        "Tags": {
          "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
          "traits": {
            "smithy.api#documentation": "<p>The tags for the specified resource.</p>"
          }
        }
      },
      "traits": {
        "smithy.api#output": {}
      }
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociations": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      }
    ]
  }
}

```

```

    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Lists user associations for an identity provider.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/user/ListUserAssociations"
    },
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "InstanceUserSummaries"
    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#ListUserAssociationsRequest": {
  "type": "structure",
  "members": {
    "InstanceId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the EC2 instance, which provides user-based
subscriptions.</p>",
        "smithy.api#required": {}
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "MaxResults": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#BoxInteger",

        "traits": {
            "smithy.api#documentation": "<p>The maximum number of results to return from a single
request.</p>"
        },
        "Filters": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>You can use the following filters to streamline results:</p> <ul>
<li> <p>Status</p> </li> <li> <p>Username</p> </li> <li> <p>Domain</p> </li> </ul>"
            }
        },
        "NextToken": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>A token to specify where to start paginating. This is the nextToken
from a previously truncated response.</p>"
            }
        },
        "traits": {
            "smithy.api#input": {}
        }
    },

    "com.amazonaws.licensemanagerusersubscriptions#ListUserAssociationsResponse": {
        "type": "structure",
        "members": {
            "InstanceUserSummaries": {
                "target": "com.amazonaws.licensemanagerusersubscriptions#InstanceUserSummaryList",
                "traits": {
                    "smithy.api#documentation": "<p>Metadata that describes the list user association operation.</p>"
                }
            },
            "NextToken": {
                "target": "smithy.api#String",
                "traits": {
                    "smithy.api#documentation": "<p>The next token used for paginated responses. When this field isn't
empty, there are additional elements that the service hasn't included in this request. Use this token with the next
request to retrieve additional objects.</p>"
                }
            }
        },
        "traits": {
            "smithy.api#output": {}
        }
    }
}

```

```

    {}
  },
  "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary": {
    "type": "structure",
    "members": {
      "Username": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The user name from the identity provider for this product
user.</p>",
          "smithy.api#required": {}
        }
      },
      "Product": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the user-based subscription product.</p>",
          "smithy.api#required": {}
        }
      },
      "IdentityProvider": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
        "traits": {
          "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
          "smithy.api#required": {}
        }
      },
      "Status": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status of a product for this user.</p>",
          "smithy.api#required": {}
        }
      },
      "ProductUserArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for this product user.</p>"
        }
      },
      "StatusMessage": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The status message for a
product for this user.</p>"
        }
      }
    }
  }
}

```

```

    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user
information for the product subscription.</p>"
      }
    },
    "SubscriptionStartDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The start date of a subscription.</p>"
      }
    },
    "SubscriptionEndDate": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The end date of a subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>A summary
of the user-based subscription products for a specific user.</p>"
    },
    "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummaryList": {
      "type": "list",
      "member": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary"
      }
    },
    "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings": {
      "type": "structure",
      "members": {
        "RdsSalCredentialsProvider": {
          "target": "com.amazonaws.licenseanagerusersubscriptions#CredentialsProvider",
          "traits": {
            "smithy.api#documentation": "<p>The <code>CredentialsProvider</code> resource contains a
reference to the credentials provider that's used for RDS license server user administration.</p>",
            "smithy.api#required": {}
          }
        }
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>Server settings that are specific to a Remote Desktop Services (RDS)
license server.</p>"
    }
  }

```

```

    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProvider": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Registers an identity provider for user-based subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/identity-provider/RegisterIdentityProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#RegisterIdentityProviderRequest":
{
  "type": "structure",
  "members": {
    "IdentityProvider": {

```

```

        "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProvider",
        "traits": {
            "smithy.api#documentation": "<p>An object that specifies details for the identity provider to
register.</p>",
            "smithy.api#required": {}
        }
    },
    "Product": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
            "smithy.api#required": {}
        }
    },
    "Settings": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Settings",
        "traits": {
            "smithy.api#documentation": "<p>The registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        }
    },
    "Tags": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#Tags",
        "traits": {
            "smithy.api#documentation": "<p>The tags that apply to the identity provider's registration.</p>"
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerusersubscriptions#RegisterIdentityProviderResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {

            "target": "com.amazonaws.licensemanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#documentation": "<p>Metadata that describes the results of an identity provider
operation.</p>",
                "smithy.api#required": {}
            }
        }
    },

```



```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceArn": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^arn:([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,63}):([a-z0-9-\\.]{1,510})/([a-z0-9-\\.]{1,510})$"
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException": {
    "type": "structure",
    "members": {
      "message": {
        "target": "smithy.api#String"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The resource couldn't be found.</p>",
    "smithy.api#error": "client",
    "smithy.api#httpError": 404
  }
},
"com.amazonaws.licensemanagerusersubscriptions#SecretsManagerCredentialsProvider": {
  "type": "structure",
  "members": {
    "SecretId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The ID of the Secrets Manager secret that contains credentials.</p>",
        "smithy.api#length": {
          "min": 1
        }
      }
    }
  }
},
"traits": {
  "smithy.api#documentation": "<p>Contains a credentials secret that's stored in Secrets Manager.</p>"
}
},
"com.amazonaws.licensemanagerusersubscriptions#SecurityGroup":
{
  "type": "string",
  "traits": {
    "smithy.api#length": {

```

```

        "min": 5,
        "max": 200
    },
    "smithy.api#pattern": "^sg-([0-9a-z]{8})([0-9a-z]{17}))$"
}
},
"com.amazonaws.licenseanagerusersubscriptions#ServerEndpoint": {
    "type": "structure",
    "members": {
        "Endpoint": {
            "target": "smithy.api#String",
            "traits": {
                "smithy.api#documentation": "<p>The network address of the endpoint.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>A network endpoint through which you can access one or more
servers.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerSettings": {
    "type": "union",

    "members": {
        "RdsSalSettings": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#RdsSalSettings",
            "traits": {
                "smithy.api#documentation": "<p>The <code>RdsSalSettings</code> resource contains settings to
configure a specific Remote Desktop Services (RDS) license server.</p>"
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Contains settings for a specific server.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#ServerType": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "RDS_SAL",
                "value": "RDS_SAL",
                "documentation": ""
            }
        ]
    }
}

```

```

    },
    "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException":
    {
        "type": "structure",
        "members": {
            "message": {
                "target": "smithy.api#String"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>The request failed because a service quota is exceeded.</p>",
            "smithy.api#error": "client"
        }
    },
    "com.amazonaws.licenseanagerusersubscriptions#Settings": {
        "type": "structure",
        "members": {
            "Subnets": {
                "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",
                "traits": {
                    "smithy.api#documentation": "<p>The subnets defined for the registered identity provider.</p>",
                    "smithy.api#length": {
                        "min": 1
                    },
                    "smithy.api#required": {}
                }
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688 communication between resources in your VPC and the VPC endpoint for activation servers.</p>",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The registered identity providers product related configuration settings such as the subnets to provision VPC endpoints, and the security group ID that is associated with the VPC endpoints. The security group should permit inbound TCP port 1688 communication from resources in the VPC.</p>"
    }
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscription": {
    "type": "operation",
    "input":

```

```

{
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest"
},
"output": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse"
},
"errors": [
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
  },
  {
    "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
  }
],
"traits": {
  "smithy.api#documentation": "<p>Starts a product subscription for a user with the specified identity provider.</p> <note> <p>Your estimated bill for charges on the number of users and related costs will take 48 hours to appear for billing periods that haven't closed (marked as <b>Pending</b> billing status) in Amazon Web Services Billing. For more information, see <a href='\"https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/invoice.html\"'>Viewing your monthly charges</a> in the <i>Amazon Web Services Billing User Guide</i>.</p> </note>",
  "smithy.api#http": {
    "code": 200,
    "method": "POST",
  },
  "uri": "/user/StartProductSubscription"
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionRequest": {
  "type": "structure",
  "members": {

```

```

"Username": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The user name from the identity provider of the user.</p>",
    "smithy.api#required": {}
  }
},
"IdentityProvider": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
  "traits": {
    "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>",
    "smithy.api#required": {}
  }
},
"Product": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid
values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>",
    "smithy.api#required": {}
  }
},
"Domain": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for
whom to start the product subscription.</p>"
  }
},
"Tags": {
  "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
  "traits": {
    "smithy.api#documentation": "<p>The tags that apply to the product subscription.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StartProductSubscriptionResponse": {
  "type": "structure",
  "members": {
    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {

```

```

        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
    }
},
"traits": {
    "smithy.api#output": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscription": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest"
    },
    "output": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse"
    },
    "errors": [
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ConflictException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ServiceQuotaExceededException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
        },
        {
            "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
        }
    ],
    "traits": {
        "smithy.api#documentation": "<p>Stops a product subscription for a user with the specified identity
provider.</p>",
        "smithy.api#http": {
            "code": 200,
            "method": "POST",
            "uri": "/user/StopProductSubscription"
        }
    }
}

```

```

    }
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionRequest": {
  "type": "structure",
  "members": {
    "Username": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The user name from the identity provider for the user.</p>"
      }
    },
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider",
      "traits": {
        "smithy.api#documentation": "<p>An object that specifies details for the identity provider.</p>"
      }
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> | <code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
      }
    },
    "ProductUserArn": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the product user.</p>"
      }
    },
    "Domain": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The domain name of the Active Directory that contains the user for whom to stop the product subscription.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#StopProductSubscriptionResponse": {
  "type": "structure",
  "members": {

```

```

    "ProductUserSummary": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#ProductUserSummary",
      "traits": {
        "smithy.api#documentation": "<p>Metadata that describes the start product subscription
operation.</p>",
        "smithy.api#required": {}
      }
    },
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnet": {
    "type": "string",
    "traits": {
      "smithy.api#pattern": "^subnet-[a-z0-9]{8,17}"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#Subnets": {
    "type": "list",
    "member": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#Subnet"
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagKeyList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResource": {
    "type": "operation",

```



```

    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#TagResourceResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":
        "<p>Adds tags to a resource.</p>",
      "smithy.api#http": {
        "method": "PUT",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#TagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to tag.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      },
      "Tags": {
        "target": "com.amazonaws.licenseanagerusersubscriptions#Tags",
        "traits": {
          "smithy.api#documentation":
            "<p>The tags to apply to the specified resource.</p>",
          "smithy.api#required": {}
        }
      }
    }
  },

```

```

    "traits": {
      "smithy.api#input": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#TagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#Tags": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    },
    "traits": {
      "smithy.api#length": {
        "min": 0,
        "max": 50
      },
      "smithy.api#sensitive": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#ThrottlingException":
{
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied because of request throttling. Retry the request.</p>",
    "smithy.api#error": "client"
  }
},
  "com.amazonaws.licensemanagerusersubscriptions#UntagResource": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse"
    }
  }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#InternalServerError"
      },
      {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceNotFoundException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Removes tags from a resource.</p>",
      "smithy.api#http": {
        "method": "DELETE",
        "uri": "/tags/{ResourceArn}"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerusersubscriptions#UntagResourceRequest": {
    "type": "structure",
    "members": {
      "ResourceArn": {
        "target": "com.amazonaws.licensemanagerusersubscriptions#ResourceArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource that you want
to remove tags from.</p>",
          "smithy.api#httpLabel": {},
          "smithy.api#required": {}
        }
      }
    },
    "TagKeys": {
      "target": "com.amazonaws.licensemanagerusersubscriptions#TagKeyList",
      "traits": {
        "smithy.api#documentation": "<p>The tag keys to remove from the resource.</p>",
        "smithy.api#httpQuery": "tagKeys",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerusersubscriptions#UntagResourceResponse": {
  "type": "structure",
  "members": {},

```

```

    "traits": {
      "smithy.api#output": {}
    }
  },
  "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettings": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#AccessDeniedException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licenseanagerusersubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates additional product configuration settings for the registered identity provider.</p>",
      "smithy.api#http": {

"code": 200,
      "method": "POST",
      "uri": "/identity-provider/UpdateIdentityProviderSettings"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsRequest": {
  "type": "structure",
  "members": {
    "IdentityProvider": {
      "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProvider"
    },
    "Product": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The name of the user-based subscription product.</p> <p>Valid

```

```

values: <code>VISUAL_STUDIO_ENTERPRISE</code> | <code>VISUAL_STUDIO_PROFESSIONAL</code> |
<code>OFFICE_PROFESSIONAL_PLUS</code> | <code>REMOTE_DESKTOP_SERVICES</code> </p>"
    }
  },
  "IdentityProviderArn":
{
    "target": "com.amazonaws.licenseanagerusersubscriptions#Arn",
    "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the identity provider to
update.</p>"
    }
  },
  "UpdateSettings": {
    "target": "com.amazonaws.licenseanagerusersubscriptions#UpdateSettings",
    "traits": {
        "smithy.api#documentation": "<p>Updates the registered identity providers product related
configuration settings. You can update any combination of settings in a single operation such as the:</p> <ul> <li>
<p>Subnets which you want to add to provision VPC endpoints.</p> </li> <li> <p>Subnets which you want to
remove the VPC endpoints from.</p> </li> <li> <p>Security group ID which permits traffic to the VPC
endpoints.</p> </li> </ul>",
        "smithy.api#required": {}
    }
  }
},
"traits": {
    "smithy.api#input": {}
}
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateIdentityProviderSettingsResponse": {
    "type": "structure",
    "members": {
        "IdentityProviderSummary": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#IdentityProviderSummary",
            "traits": {
                "smithy.api#required": {}
            }
        }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licenseanagerusersubscriptions#UpdateSettings": {
    "type": "structure",
    "members": {
        "AddSubnets": {
            "target": "com.amazonaws.licenseanagerusersubscriptions#Subnets",

```

```

        "traits": {
            "smithy.api#documentation": "<p>The ID of one
or more subnets in which License Manager will create a VPC endpoint for products that require connectivity to
activation servers.</p>",
            "smithy.api#required": {}
        },
        "RemoveSubnets": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#Subnets",
            "traits": {
                "smithy.api#documentation": "<p>The ID of one or more subnets to remove.</p>",
                "smithy.api#required": {}
            }
        },
        "SecurityGroupId": {
            "target": "com.amazonaws.licensemanagerusersubscriptions#SecurityGroup",
            "traits": {
                "smithy.api#documentation": "<p>A security group ID that allows inbound TCP port 1688
communication between resources in your VPC and the VPC endpoints for activation servers.</p>"
            }
        },
        "traits": {
            "smithy.api#documentation": "<p>Updates the registered identity providers product related configuration
settings such as the subnets to provision VPC endpoints.</p>"
        },
        "com.amazonaws.licensemanagerusersubscriptions#ValidationException": {
            "type": "structure",
            "members": {
                "message": {
                    "target": "smithy.api#String"
                }
            },
            "traits": {
                "smithy.api#documentation": "<p>A parameter is not valid.</p>",
                "smithy.api#error": "client"
            }
        }
    }
}

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

import (
    "context"

```

```

"fmt"
awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
"github.com/aws/aws-sdk-go-v2/service/grafana/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Removes the Grafana Enterprise license from a workspace.
func (c *Client) DisassociateLicense(ctx context.Context, params *DisassociateLicenseInput, optFns
...func(*Options)) (*DisassociateLicenseOutput, error) {
    if params == nil {
        params = &DisassociateLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "DisassociateLicense", params, optFns,
c.addOperationDisassociateLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DisassociateLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DisassociateLicenseInput struct {

    // The type of license to remove from the workspace.
    //
    // This member is required.
    LicenseType types.LicenseType

    //
    // The ID of the workspace to remove the Grafana Enterprise license from.
    //
    // This member is required.
    WorkspaceId *string

    noSmithyDocumentSerde
}

type DisassociateLicenseOutput struct {

    // A structure containing information about the workspace.
    //
    // This member is required.
    Workspace *types.WorkspaceDescription

```

```

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationDisassociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsRestjson1_serializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsRestjson1_deserializeOpDisassociateLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "DisassociateLicense");
    err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {
        return err
    }
    if err = addSetLoggerMiddleware(stack, options); err != nil {
        return err
    }
    if err = addClientRequestID(stack); err != nil {
        return err
    }
    if err = addComputeContentLength(stack); err != nil {
        return err
    }
    if err = addResolveEndpointMiddleware(stack, options); err != nil {
        return err
    }
    if err = addComputePayloadSHA256(stack); err != nil {
        return err
    }
    if err = addRetry(stack, options); err != nil {
        return err
    }
    if err = addRawResponseToMetadata(stack); err != nil {
        return err
    }
}

```



```

if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if
err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDisassociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDisassociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options);

```

```

err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if
err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDisassociateLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DisassociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

package licensemanager

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Gets detailed information about the specified license.
func (c *Client) GetLicense(ctx context.Context, params *GetLicenseInput, optFns ...func(*Options))
(*GetLicenseOutput, error) {
    if params == nil {
        params = &GetLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "GetLicense", params, optFns,
c.addOperationGetLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*GetLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type GetLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // License version.

```

Version *string

noSmithyDocumentSerde
}

type

GetLicenseOutput struct {

// License details.

License *types.License

// Metadata pertaining to the operation's result.

ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

```
func (c *Client) addOperationGetLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {  
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {  
        return err  
    }  
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpGetLicense{}, middleware.After)  
    if err != nil {  
        return err  
    }  
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpGetLicense{}, middleware.After)  
    if err != nil {  
        return err  
    }  
    if err := addProtocolFinalizerMiddlewares(stack, options, "GetLicense"); err != nil {  
        return fmt.Errorf("add protocol finalizers: %v", err)  
    }  
}
```

```
if err = addLegacyEndpointContextSetter(stack, options); err != nil {  
    return err  
}
```

```
if err = addSetLoggerMiddleware(stack, options); err != nil {  
    return err  
}
```

```
if err = addClientRequestID(stack); err  
!= nil {  
    return err  
}
```

```
if err = addComputeContentLength(stack); err != nil {  
    return err  
}
```

```
if err = addResolveEndpointMiddleware(stack, options); err != nil {  
    return err  
}
```

```

}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options);
err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpGetLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opGetLicense(options.Region), middleware.Before);
err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
}

```

```

if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack,
options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}

```

```

}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

func newServiceMetadataMiddleware_opGetLicense(region string) *awsmiddleware.RegisterServiceMetadata {
    return &awsmiddleware.RegisterServiceMetadata{
        Region:
            region,
        ServiceID:  ServiceID,
        OperationName: "GetLicense",
    }
}
name: License Scan

on: [pull_request]

permissions:
    contents: read

jobs:
    licensescan:
        name: License Scan
        runs-on: ubuntu-latest
        strategy:
            matrix:
                python-version: [3.9]

        steps:
            - name: Checkout target
              uses: actions/checkout@v2
              with:
                path: sdkbase
                ref: ${{ github.base_ref }}
            - name: Checkout this ref
              uses: actions/checkout@v2
              with:
                path: new-ref
                fetch-depth: 0
            - name: Get Diff
              run: git --git-dir ./new-ref/.git diff --name-only --diff-filter=ACMRT ${{ github.event.pull_request.base.sha }}
                ${{ github.sha }} > refDiffFiles.txt

```

```

- name: Get Target Files
  run: git --git-dir ./sdkbase/.git ls-files | grep -xf refDiffFiles.txt -> targetFiles.txt
- name: Checkout scancode
  uses: actions/checkout@v2
  with:
    repository: nexB/scancode-toolkit
    path: scancode-toolkit
    fetch-depth: 1
- name: Set up Python
  ${{ matrix.python-version }}
  uses: actions/setup-python@v2
  with:
    python-version: ${{ matrix.python-version }}
# ScanCode
- name: Self-configure scancode
  working-directory: ./scancode-toolkit
  run: ./scancode --help
- name: Run Scan code on target
  run: cat targetFiles.txt | while read filename; do echo ./sdkbase/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> old-licenses.txt
- name: Run Scan code on pr ref
  run: cat refDiffFiles.txt | while read filename; do echo ./new-ref/$filename; done | xargs ./scancode-
toolkit/scancode -l -n 30 --json-pp - | grep short_name | sort | uniq >> new-licenses.txt
# compare
- name: License test
  run: if ! cmp old-licenses.txt new-licenses.txt; then echo "Licenses differ! Failing."; exit -1; else echo "Licenses
are the same. Success."; exit 0; fi
// Code generated by smithy-go-codegen DO NOT EDIT.

```

package licensemanager

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license.
//
// If the account that created the license is the same that is performing the
// check out, you must specify the account as the beneficiary.
func (c *Client) CheckoutLicense(ctx context.Context, params *CheckoutLicenseInput, optFns ...func(*Options))
(*CheckoutLicenseOutput, error) {
    if params == nil {
        params = &CheckoutLicenseInput{}
    }
}

```



```

}

result, metadata, err := c.invokeOperation(ctx, "CheckoutLicense", params, optFns,
c.addOperationCheckoutLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckoutLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckoutLicenseInput struct {

    //
    Checkout type.
    //
    // This member is required.
    CheckoutType types.CheckoutType

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    // the request.
    //
    // This member is required.
    ClientToken *string

    // License entitlements.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Key fingerprint identifying the license.
    //
    // This member is required.
    KeyFingerprint *string

    // Product SKU.
    //
    // This member is required.
    ProductSKU *string

    // License beneficiary.
    Beneficiary *string

    // Node ID.
    NodeId *string

```

```

noSmithyDocumentSerde
}

type CheckoutLicenseOutput struct {

    // Checkout type.
    CheckoutType types.CheckoutType

    // Allowed license entitlements.
    EntitlementsAllowed []types.EntitlementData

    // Date and time at which the license checkout expires.
    Expiration *string

    // Date and time at which the license checkout is issued.
    IssuedAt *string

    // Amazon Resource Name (ARN) of the checkout license.
    LicenseArn
    *string

    // License consumption token.
    LicenseConsumptionToken *string

    // Node ID.
    NodeId *string

    // Signed token.
    SignedToken *string

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutLicense{}, middleware.After)
    if err != nil {
        return err
    }
}

```

```

if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack,
options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,

```

```

c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutLicense",
}
}

```

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"

```

```

"github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
"github.com/aws/smithy-go/middleware"
smithyhttp "github.com/aws/smithy-go/transport/http"
)

// Checks out the specified license for offline use.
func (c *Client) CheckoutBorrowLicense(ctx context.Context, params *CheckoutBorrowLicenseInput, optFns
...func(*Options)) (*CheckoutBorrowLicenseOutput, error) {
    if params == nil {
        params = &CheckoutBorrowLicenseInput{}
    }

    result, metadata, err := c.invokeOperation(ctx, "CheckoutBorrowLicense", params, optFns,
c.addOperationCheckoutBorrowLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*CheckoutBorrowLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type CheckoutBorrowLicenseInput struct {

    // Unique, case-sensitive identifier that you provide to ensure the idempotency of
    //
    the request.
    //
    // This member is required.
    ClientToken *string

    // Digital signature method. The possible value is JSON Web Signature (JWS)
    // algorithm PS384. For more information, see [RFC 7518 Digital Signature with RSASSA-PSS].
    //
    // [RFC 7518 Digital Signature with RSASSA-PSS]: https://tools.ietf.org/html/rfc7518#section-3.5
    //
    // This member is required.
    DigitalSignatureMethod types.DigitalSignatureMethod

    // License entitlements. Partial checkouts are not supported.
    //
    // This member is required.
    Entitlements []types.EntitlementData

    // Amazon Resource Name (ARN) of the license. The license must use the borrow
    // consumption configuration.
    //

```

```

// This member is required.
LicenseArn *string

// Information about constraints.
CheckoutMetadata []types.Metadata

// Node ID.
NodeId *string

noSmithyDocumentSerde
}

type CheckoutBorrowLicenseOutput struct {

// Information about constraints.
CheckoutMetadata []types.Metadata

// Allowed license entitlements.
EntitlementsAllowed
[]types.EntitlementData

// Date and time at which the license checkout expires.
Expiration *string

// Date and time at which the license checkout is issued.
IssuedAt *string

// Amazon Resource Name (ARN) of the license.
LicenseArn *string

// License consumption token.
LicenseConsumptionToken *string

// Node ID.
NodeId *string

// Signed token.
SignedToken *string

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationCheckoutBorrowLicenseMiddlewares(stack *middleware.Stack, options Options) (err
error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{ }, middleware.After); err != nil {

```

```

    return err
}
err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckoutBorrowLicense{}, middleware.After)
if err != nil {
    return
err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "CheckoutBorrowLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
}

```



```

if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack);
err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckoutBorrowLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckoutBorrowLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options);
err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return
err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckoutBorrowLicense(region string)
*awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "CheckoutBorrowLicense",
}
}

```

// Code generated by smithy-go-codegen DO NOT EDIT.

package grafana

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/aws-sdk-go-v2/service/grafana/types"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Assigns a Grafana Enterprise license to a workspace. To upgrade, you must use  
// ENTERPRISE for the licenseType , and pass in a valid Grafana Labs token for the  
// grafanaToken . Upgrading to Grafana Enterprise incurs additional fees. For more  
// information, see [Upgrade a workspace to Grafana Enterprise].  
//
```

```
// [Upgrade a workspace to Grafana Enterprise]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-Grafana-Enterprise.html
```

```
func (c *Client) AssociateLicense(ctx context.Context, params *AssociateLicenseInput, optFns ...func(*Options))  
(*AssociateLicenseOutput, error) {  
    if params == nil {  
        params = &AssociateLicenseInput{}
```

```
    }  
  
    result, metadata, err := c.invokeOperation(ctx,  
        "AssociateLicense", params, optFns, c.addOperationAssociateLicenseMiddlewares)  
    if err != nil {  
        return nil, err  
    }
```

```
  
    out := result.(*AssociateLicenseOutput)  
    out.ResultMetadata = metadata  
    return out, nil  
}
```

```
type AssociateLicenseInput struct {
```

```
    // The type of license to associate with the workspace.
```

```
    //
```

```
    // Amazon Managed Grafana workspaces no longer support Grafana Enterprise free  
    // trials.
```

```
    //
```

```
    // This member is required.
```

```
    LicenseType types.LicenseType
```

```
    // The ID of the workspace to associate the license with.
```

```

//
// This member is required.
WorkspaceId *string

// A token from Grafana Labs that ties your Amazon Web Services account with a
// Grafana Labs account. For more information, see [Link your account with Grafana Labs].
//
// [Link your account with Grafana Labs]: https://docs.aws.amazon.com/grafana/latest/userguide/upgrade-to-
Grafana-Enterprise.html#AMG-workspace-register-enterprise
GrafanaToken *string

noSmithyDocumentSerde
}

type AssociateLicenseOutput
struct {

// A structure containing data about the workspace.
//
// This member is required.
Workspace *types.WorkspaceDescription

// Metadata pertaining to the operation's result.
ResultMetadata middleware.Metadata

noSmithyDocumentSerde
}

func (c *Client) addOperationAssociateLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
return err
}
err = stack.Serialize.Add(&awsRestjson1_serializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
err = stack.Deserialize.Add(&awsRestjson1_deserializeOpAssociateLicense{}, middleware.After)
if err != nil {
return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "AssociateLicense"); err != nil {
return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err != nil {
return err
}
if err = addSetLoggerMiddleware(stack,

```

```

options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack,
c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}

```

```

}
if err = addOpAssociateLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opAssociateLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack,
options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
}

```

```

if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opAssociateLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "AssociateLicense",
}
}
// Code generated by smithy-go-codegen DO NOT EDIT.

```

```

package licensemanager

```

```

import (
    "context"
    "fmt"
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"
    "github.com/aws/aws-sdk-go-v2/service/licensemanager/types"
    "github.com/aws/smithy-go/middleware"
    smithyhttp "github.com/aws/smithy-go/transport/http"
)

```

```

// Deletes the specified license.

```

```

func (c *Client) DeleteLicense(ctx context.Context, params *DeleteLicenseInput, optFns ...func(*Options))
(*DeleteLicenseOutput, error) {
    if params == nil {
        params = &DeleteLicenseInput{}
    }
}

```

```

    result, metadata, err := c.invokeOperation(ctx, "DeleteLicense", params, optFns,
c.addOperationDeleteLicenseMiddlewares)
    if err != nil {
        return nil, err
    }

    out := result.(*DeleteLicenseOutput)
    out.ResultMetadata = metadata
    return out, nil
}

type DeleteLicenseInput struct {

    // Amazon Resource Name (ARN) of the license.
    //
    // This member is required.
    LicenseArn *string

    // Current version of the license.
    //
    // This member is required.
    SourceVersion
    *string

    noSmithyDocumentSerde
}

type DeleteLicenseOutput struct {

    // Date when the license is deleted.
    DeletionDate *string

    // License status.
    Status types.LicenseDeletionStatus

    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationDeleteLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpDeleteLicense{}, middleware.After)
    if err != nil {

```



```

    return err
}
err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpDeleteLicense{}, middleware.After)
if err != nil {
    return err
}
if err := addProtocolFinalizerMiddlewares(stack, options, "DeleteLicense"); err != nil {
    return fmt.Errorf("add protocol finalizers: %v", err)
}

if err = addLegacyEndpointContextSetter(stack, options); err !=
nil {
    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {

```

```

    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err
!= nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
if err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpDeleteLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opDeleteLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err !=
nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err != nil {

```

```

    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opDeleteLicense(region
string) *awsmiddleware.RegisterServiceMetadata {
return &awsmiddleware.RegisterServiceMetadata{
    Region:    region,
    ServiceID: ServiceID,
    OperationName: "DeleteLicense",
}
}
CheckInLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata

```

legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Code generated by smithy-go-codegen DO NOT EDIT.

```
package licensemanager
```

```
import (  
    "context"  
    "fmt"  
    awsmiddleware "github.com/aws/aws-sdk-go-v2/aws/middleware"  
    "github.com/aws/smithy-go/middleware"  
    smithyhttp "github.com/aws/smithy-go/transport/http"  
)
```

```
// Checks in the specified license. Check in a license when it is no longer in use.
```

```
func (c *Client) CheckInLicense(ctx context.Context, params *CheckInLicenseInput, optFns ...func(*Options))  
(*CheckInLicenseOutput, error) {  
    if params == nil {  
        params = &CheckInLicenseInput{}  
    }  
}
```

```
result, metadata, err := c.invokeOperation(ctx, "CheckInLicense", params, optFns,
```

```

c.addOperationCheckInLicenseMiddlewares)
if err != nil {
    return nil, err
}

out := result.(*CheckInLicenseOutput)
out.ResultMetadata = metadata
return out, nil
}

type CheckInLicenseInput struct {

    // License consumption token.
    //
    // This member is required.
    LicenseConsumptionToken *string

    // License beneficiary.
    Beneficiary *string

    noSmithyDocumentSerde
}

type
CheckInLicenseOutput struct {
    // Metadata pertaining to the operation's result.
    ResultMetadata middleware.Metadata

    noSmithyDocumentSerde
}

func (c *Client) addOperationCheckInLicenseMiddlewares(stack *middleware.Stack, options Options) (err error) {
    if err := stack.Serialize.Add(&setOperationInputMiddleware{}, middleware.After); err != nil {
        return err
    }
    err = stack.Serialize.Add(&awsAwsjson11_serializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    err = stack.Deserialize.Add(&awsAwsjson11_deserializeOpCheckInLicense{}, middleware.After)
    if err != nil {
        return err
    }
    if err := addProtocolFinalizerMiddlewares(stack, options, "CheckInLicense"); err != nil {
        return fmt.Errorf("add protocol finalizers: %v", err)
    }

    if err = addLegacyEndpointContextSetter(stack, options); err != nil {

```

```

    return err
}
if err = addSetLoggerMiddleware(stack, options); err != nil {
    return err
}
if err = addClientRequestID(stack); err != nil {
    return err
}
if
err = addComputeContentLength(stack); err != nil {
    return err
}
if err = addResolveEndpointMiddleware(stack, options); err != nil {
    return err
}
if err = addComputePayloadSHA256(stack); err != nil {
    return err
}
if err = addRetry(stack, options); err != nil {
    return err
}
if err = addRawResponseToMetadata(stack); err != nil {
    return err
}
if err = addRecordResponseTiming(stack); err != nil {
    return err
}
if err = addSpanRetryLoop(stack, options); err != nil {
    return err
}
if err = addClientUserAgent(stack, options); err != nil {
    return err
}
if err = smithyhttp.AddErrorCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = smithyhttp.AddCloseResponseBodyMiddleware(stack); err != nil {
    return err
}
if err = addSetLegacyContextSigningOptionsMiddleware(stack); err != nil {
    return err
}
if err = addTimeOffsetBuild(stack, c); err != nil {
    return err
}
if err = addUserAgentRetryMode(stack, options); err != nil {
    return err
}
}

```

```

if
err = addCredentialSource(stack, options); err != nil {
    return err
}
if err = addOpCheckInLicenseValidationMiddleware(stack); err != nil {
    return err
}
if err = stack.Initialize.Add(newServiceMetadataMiddleware_opCheckInLicense(options.Region),
middleware.Before); err != nil {
    return err
}
if err = addRecursionDetection(stack); err != nil {
    return err
}
if err = addRequestIDRetrieverMiddleware(stack); err != nil {
    return err
}
if err = addResponseErrorMiddleware(stack); err != nil {
    return err
}
if err = addRequestResponseLogging(stack, options); err != nil {
    return err
}
if err = addDisableHTTPSMiddleware(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeRetryLoop(stack, options); err != nil {
    return err
}
if err = addInterceptAttempt(stack, options); err != nil {
    return err
}
if err = addInterceptExecution(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSerialization(stack, options); err !=
nil {
    return err
}
if err = addInterceptAfterSerialization(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeSigning(stack, options); err != nil {
    return err
}
if err = addInterceptAfterSigning(stack, options); err != nil {
    return err
}
}

```



```

if err = addInterceptTransmit(stack, options); err != nil {
    return err
}
if err = addInterceptBeforeDeserialization(stack, options); err != nil {
    return err
}
if err = addInterceptAfterDeserialization(stack, options); err != nil {
    return err
}
if err = addSpanInitializeStart(stack); err != nil {
    return err
}
if err = addSpanInitializeEnd(stack); err != nil {
    return err
}
if err = addSpanBuildRequestStart(stack); err != nil {
    return err
}
if err = addSpanBuildRequestEnd(stack); err != nil {
    return err
}
return nil
}

```

```

func newServiceMetadataMiddleware_opCheckInLicense(region string) *awsmiddleware.RegisterServiceMetadata
{
    return &awsmiddleware.RegisterServiceMetadata{
        Region:    region,
        ServiceID:
            ServiceID,
        OperationName: "CheckInLicense",
    }
}
CreateLicense
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
OperationInputValidation
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization
OperationSerializer

```

InterceptAfterSerialization
 Build stack step
 ClientRequestID
 ComputeContentLength
 SetCredentialSourceMiddleware
 UserAgent
 AddTimeOffsetMiddleware
 RecursionDetection
 spanBuildRequestEnd
 Finalize stack step
 spanRetryLoop
 InterceptBeforeRetryLoop
 Retry
 InterceptAttempt
 RetryMetricsHeader
 ResolveAuthScheme
 GetIdentity
 ResolveEndpointV2
 disableHTTPS
 ComputePayloadHash
 setLegacyContextSigningOptions
 InterceptBeforeSigning
 Signing
 InterceptAfterSigning
 Deserialize stack step
 AddRawResponseToMetadata
 ErrorCloseResponseBody
 CloseResponseBody
 ResponseErrorWrapper
 RequestIDRetriever
 InterceptAfterDeserialization
 OperationDeserializer
 InterceptBeforeDeserialization
 AddTimeOffsetMiddleware
 RecordResponseTiming
 RequestResponseLogger
 InterceptTransmit
 {
 "smithy": "2.0",
 "shapes": {
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger": {
 "type": "integer"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong": {
 "type": "long"
 },
 "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider": {
 "type": "operation",

```

    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Remove a third-party subscription provider from the Bring Your Own License (BYOL) subscriptions \n\t\t\tregistered to your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/DeregisterSubscriptionProvider"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderRequest": {
    "type": "structure",
    "members": {
      "SubscriptionProviderArn": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
        "traits": {
          "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the subscription provider resource to deregister.</p>",
          "smithy.api#required": {}
        }
      }
    },
    "traits": {
      "smithy.api#input": {}
    }
  },

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {},
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Filter": {
  "type": "structure",
  "members": {
    "Name": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The type of name to filter by.</p>"
      }
    },
    "Values": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
      "traits": {
        "smithy.api#documentation": "<p>One or more values for the name to filter by.</p>"
      }
    },
    "Operator": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Operator",
      "traits": {
        "smithy.api#documentation": "<p>An operator for filtering results.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>A filter object that is used to return more specific results from a describe operation.\n    Filters can be used to match a set of resources by specific criteria.</p>"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#FilterList": {
  "type": "list",
  "member": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Filter"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest"
  },
  "output": {
    "target":

```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Get details for a Bring Your Own License (BYOL) subscription that's
registered to your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/GetRegisteredSubscriptionProvider"
    },
    "smithy.api#idempotent": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation":
"<p>The Amazon Resource Name (ARN) of the BYOL registration resource to get details for.</p>",
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",

```

```

    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) for the BYOL registration
resource specified in the request.</p>"
    },
  },
  "SubscriptionProviderSource": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",

    "traits": {
      "smithy.api#documentation": "<p>The subscription provider for the BYOL registration resource
specified \n\t\t\tin the request.</p>"
    },
  },
  "SecretArn": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
    "traits": {
      "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the third-party access
secret stored in Secrets Manager for the BYOL \n\t\t\tregistration resource specified in the request.</p>"
    },
  },
  "SubscriptionProviderStatus": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
    "traits": {
      "smithy.api#documentation": "<p>The status of the Linux subscription provider access token from the
last \n\t\t\t\t\tsuccessful subscription data request.</p>"
    },
  },
  "SubscriptionProviderStatusMessage": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The detailed message from your subscription provider token
status.</p>"
    },
  },
  "LastSuccessfulDataRetrievalTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The timestamp from the last time License Manager retrieved
subscription details \n\t\t\t\t\tfrom your registered third-party Linux subscription provider.</p>"
    },
  },
  "traits": {
    "smithy.api#output": {}
  },
},
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings": {
  "type": "operation",

```

```

    "input": {
      "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Lists the Linux subscriptions service settings for your account.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/GetServiceSettings"
      },
      "smithy.api#idempotent": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsRequest": {
      "type": "structure",
      "members": {}
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettingsResponse": {
      "type": "structure",
      "members": {
        "LinuxSubscriptionsDiscovery": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
          "traits": {
            "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
          }
        },
        "LinuxSubscriptionsDiscoverySettings": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
          "traits": {
            "smithy.api#documentation": "<p>Lists
the settings defined for Linux subscriptions discovery. The settings include ifn    Organizations integration has
been enabled, and which Regions data will be aggregated from.</p>"

```

```

    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n  subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Instance": {
  "type": "structure",
  "members": {
    "AmiId": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The AMI ID used to launch the instance.</p>"
      }
    },
    "InstanceID": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance ID of the resource.</p>"
      }
    },
    "InstanceType": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The instance type of the resource.</p>"
      }
    },
    "AccountID": {

```



```

    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The account ID which owns the instance.</p>"
    }
  },
  "Status": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The status of the instance.</p>"
    }
  },
  "Region": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The Region the instance is running in.</p>"
    }
  },
  "UsageOperation": {
    "target": "smithy.api#String",

    "traits": {
      "smithy.api#documentation": "<p>The usage operation of the instance. For more information, see For
more information, see\n      <a href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-
subscriptions-usage-operation.html\">Usage\n      operation values</a> in the <i>License Manager User
Guide</i>.</p>"
    }
  },
  "ProductCode": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList",
    "traits": {
      "smithy.api#documentation": "<p>The product code for the instance. For more information, see <a
href=\"https://docs.aws.amazon.com/license-manager/latest/userguide/linux-subscriptions-usage-
operation.html\">Usage\n      operation values</a> in the <i>License Manager User Guide</i> .</p>"
    }
  },
  "LastUpdatedTime": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The time in which the last discovery updated the instance
details.</p>"
    }
  },
  "SubscriptionName": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>The name of the license subscription that the instance uses.</p>"
    }
  },

```

```
"OsVersion": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The operating system software version that runs on your instance.</p>"
    }
},
"SubscriptionProviderCreateTime": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The timestamp when you registered the third-party Linux subscription \n\t\t\tprovider for the subscription that the instance uses.</p>"
    }
},
"SubscriptionProviderUpdateTime": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>The timestamp from the last time that the instance synced with the registered \n\t\t\tthird-party Linux subscription provider.</p>"
    }
},
"DualSubscription": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>Indicates that you have two different license subscriptions for \n\t\t\tthe same software on your instance.</p>"
    }
},
"RegisteredWithSubscriptionProvider": {
    "target": "smithy.api#String",
    "traits": {
        "smithy.api#documentation": "<p>Indicates that your instance uses a BYOL license subscription from \n\t\t\tthird-party Linux subscription provider that you've registered with License Manager.</p>"
    }
},
"traits": {
    "smithy.api#documentation": "<p>Details discovered information about a running instance using Linux subscriptions.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList": {
    "type": "list",
    "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Instance"
    }
},
```

```

"com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation":
"<p>An exception occurred with the service.</p>",
    "smithy.api#error": "server"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LicenseManagerLinuxSubscriptions": {
  "type": "service",
  "version": "2018-05-10",
  "operations": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#DeregisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetRegisteredSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#GetServiceSettings"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResource"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource"
    },
    {

```

```

    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings"
  }
],
"traits": {
  "aws.api#service": {
    "sdkId": "License Manager Linux Subscriptions",
    "arnNamespace": "license-manager-linux-subscriptions",
    "cloudTrailEventSource": "license-manager-linux-subscriptions.amazonaws.com"
  },
  "aws.auth#sigv4": {
    "name": "license-manager-linux-subscriptions"
  },
  "aws.protocols#restJson1": {},
  "smithy.api#cors": {
    "additionalAllowedHeaders": [
      "*",
      "content-type",
      "x-amz-content-sha256",
      "x-amz-user-agent",
      "x-amzn-platform-id",
      "x-amzn-trace-id"
    ],
    "additionalExposedHeaders": [
      "x-amzn-errortype",
      "x-amzn-requestid",
      "x-amzn-trace-id"
    ]
  },
  "smithy.api#documentation": "<p>With License Manager,  
you can discover and track your commercial Linux subscriptions on running Amazon EC2 instances.</p>",
  "smithy.api#title": "AWS License Manager Linux Subscriptions",
  "smithy.rules#endpointRuleSet": {
    "version": "1.0",
    "parameters": {
      "Region": {
        "builtIn": "AWS::Region",
        "required": false,
        "documentation": "The AWS region used to dispatch the request.",
        "type": "String"
      },
      "UseDualStack": {
        "builtIn": "AWS::UseDualStack",
        "required": true,
        "default": false,
        "documentation": "When true, use the dual-stack endpoint. If the configured endpoint does not  
support dual-stack, dispatching the request MAY return an error.",
        "type": "Boolean"
      }
    }
  }
}

```

```

    "UseFIPS": {
      "builtIn": "AWS::UseFIPS",
      "required": true,
      "default": false,
      "documentation": "When true, send this request to the FIPS-compliant regional endpoint. If the
configured endpoint does not have a FIPS compliant endpoint, dispatching the request will return an error.",
      "type": "Boolean"
    },
    "Endpoint": {
      "builtIn": "SDK::Endpoint",
      "required": false,
      "documentation": "Override the endpoint used to send this request",
      "type": "String"
    }
  },
  "rules": [
    {
      "conditions": [
        {
          "fn": "isSet",
          "argv": [
            {
              "ref": "Endpoint"
            }
          ]
        }
      ]
    }
  ],
  "rules": [
    {
      "conditions": [
        {
          "fn": "booleanEquals",
          "argv": [
            {
              "ref": "UseFIPS"
            },
            true
          ]
        }
      ]
    }
  ],
  "error": "Invalid Configuration: FIPS and custom endpoint are not supported",
  "type": "error"
},
{
  "conditions": [],
  "rules": [
    {

```

```

        "conditions": [
            {
                "fn": "booleanEquals",
                "argv": [
                    {
                        "ref": "UseDualStack"
                    },
                    true
                ]
            }
        ],
        "error": "Invalid Configuration: Dualstack and custom endpoint are not supported",
        "type": "error"
    },
    {
        "conditions": [],
        "endpoint": {
            "url": {
                "ref": "Endpoint"
            },
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],
"type": "tree"

}
],
"type": "tree"
},
{
    "conditions": [],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "isSet",
                    "argv": [
                        {
                            "ref": "Region"
                        }
                    ]
                }
            ]
        }
    ],
    "rules": [

```

```

    {
      "conditions": [
        {
          "fn": "aws.partition",
          "argv": [
            {
              "ref": "Region"
            }
          ],
          "assign": "PartitionResult"
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseFIPS"
                },
                true
              ]
            },
            {
              "fn": "booleanEquals",
              "argv": [
                {
                  "ref": "UseDualStack"
                },
                true
              ]
            }
          ]
        }
      ],
      "rules": [
        {
          "conditions": [
            {
              "fn": "booleanEquals",
              "argv": [
                true,
                {
                  "fn": "getAttr",
                  "argv": [
                    {

```

```

        "ref": "PartitionResult"
    },
    "supportsFIPS"
]
}
]
},
{
    "fn": "booleanEquals",
    "argv": [
        true,
        {
            "fn": "getAttr",
            "argv": [
                {
                    "ref": "PartitionResult"
                },
                "supportsDualStack"
            ]
        }
    ]
}
],
"rules": [
    {
        "conditions": [],

"rules": [
    {
        "conditions": [],
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dualStackDnsSuffix}",
            "properties": {},
            "headers": {}
        },
        "type": "endpoint"
    }
],

"type": "tree"
},
],
"type": "tree"
},
{

```



```

        "conditions": [],
        "error": "FIPS and DualStack are enabled, but this partition does not support
one or both",
        "type": "error"
    }
],
"type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseFIPS"
                },
                true
            ]
        }
    ],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        {
                            "fn": "getAttr",
                            "argv": [
                                {
                                    "ref": "PartitionResult"
                                },
                                "supportsFIPS"
                            ]
                        },
                        true
                    ]
                }
            ],
            "rules": [
                {
                    "conditions": [],
                    "rules": [
                        {
                            "conditions": [],

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-
fips.{Region}.{PartitionResult#dnsSuffix}",

        "properties": {},

            "headers": {}
        },
        "type": "endpoint"
    }
],
    "type": "tree"
}
],
    "type": "tree"
},
{
    "conditions": [],
    "error": "FIPS is enabled
but this partition does not support FIPS",
    "type": "error"
}
],
    "type": "tree"
},
{
    "conditions": [
        {
            "fn": "booleanEquals",
            "argv": [
                {
                    "ref": "UseDualStack"
                },
                true
            ]
        }
    ]
}
],
    "rules": [
        {
            "conditions": [
                {
                    "fn": "booleanEquals",
                    "argv": [
                        true,
                        {
                            "fn": "getAttr",
                            "argv": [
                                {

```

```

        "ref": "PartitionResult"
      },
      "supportsDualStack"
    ]
  }
]
}
],
"rules": [
  {
    "conditions": [],
    "rules": [
      {
        "conditions": [],
        "endpoint": {
          "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dualStackDnsSuffix}",
          "properties": {},
          "headers": {}
        },
        "type": "endpoint"
      }
    ],
    "type": "tree"
  }
],
"conditions": [],
"error": "DualStack is enabled but this partition does not support DualStack",
"type": "error"
}
],
"conditions": [],
"rules": [
  {
    "conditions": [],
    "endpoint": {
      "url": "https://license-manager-linux-
subscriptions.{Region}.{PartitionResult#dnsSuffix}",
      "properties": {},

```

```

        "headers": {}
      },
      "type": "endpoint"
    }
  ],
  "type": "tree"
}
],
"type": "tree"
}
],
"type": "tree"
},
{
  "conditions": [],
  "error": "Invalid Configuration: Missing Region",
  "type": "error"
}
],
"type": "tree"
}
]
},
"smithy.rules#endpointTests": {
  "testCases": [
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack enabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.api.aws"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": true
      }
    },
    {
      "documentation": "For region us-east-1 with FIPS enabled and DualStack disabled",
      "expect": {
        "endpoint": {
          "url": "https://license-manager-linux-subscriptions-fips.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "params": {
        "Region": "us-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region cn-north-1 with FIPS enabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.cn-north-
1.api.amazonwebservices.com.cn"
        }
    },
    "params": {
        "Region": "cn-north-1",
        "UseFIPS": true,
        "UseDualStack": true
    }
},
{

```

```

"documentation": "For region cn-north-1 with FIPS enabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": true,
    "UseDualStack": false
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.api.amazonwebservices.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": true
  }
},
{
  "documentation": "For region cn-north-1 with FIPS disabled and DualStack disabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions.cn-north-1.amazonaws.com.cn"
    }
  },
  "params": {
    "Region": "cn-north-1",
    "UseFIPS": false,
    "UseDualStack": false
  }
},
{
  "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack enabled",
  "expect": {
    "endpoint": {
      "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.api.aws"
    }
  },
  "params": {
    "Region": "us-gov-east-1",

```

```

        "UseFIPS": true,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack enabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.api.aws"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": true
    }
},
{
    "documentation": "For region us-gov-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-gov-east-1.amazonaws.com"
        }
    },
    "params": {
        "Region": "us-gov-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS enabled and DualStack disabled",
    "expect": {

```

```

        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-iso-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-iso-east-1.c2s.ic.gov"
        }
    },
    "params": {
        "Region": "us-iso-east-1",
        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS enabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions-fips.us-isob-east-1.sc2s.sgov.gov"
        }
    },
    "params": {
        "Region": "us-isob-east-1",
        "UseFIPS": true,
        "UseDualStack": false
    }
},
{
    "documentation": "For region us-isob-east-1 with FIPS disabled and DualStack disabled",
    "expect": {
        "endpoint": {
            "url": "https://license-manager-linux-subscriptions.us-isob-east-1.sc2s.sgov.gov"
        }
    }
},
    "params": {
        "Region": "us-isob-east-1",

```



```

        "UseFIPS": false,
        "UseDualStack": false
    }
},
{
    "documentation": "For custom endpoint with region set and fips disabled and dualstack disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "Region": "us-east-1",
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},

{
    "documentation": "For custom endpoint with region not set and fips disabled and dualstack
disabled",
    "expect": {
        "endpoint": {
            "url": "https://example.com"
        }
    },
    "params": {
        "UseFIPS": false,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips enabled and dualstack disabled",
    "expect": {
        "error": "Invalid Configuration: FIPS and custom endpoint are not supported"
    },
    "params": {

"Region": "us-east-1",
        "UseFIPS": true,
        "UseDualStack": false,
        "Endpoint": "https://example.com"
    }
},
{
    "documentation": "For custom endpoint with fips disabled and dualstack enabled",

```

```

        "expect": {
            "error": "Invalid Configuration: Dualstack and custom endpoint are not supported"
        },
        "params": {
            "Region": "us-east-1",
            "UseFIPS": false,
            "UseDualStack": true,
            "Endpoint": "https://example.com"
        }
    },
    {
        "documentation": "Missing region",

        "expect": {
            "error": "Invalid Configuration: Missing Region"
        }
    }
],
"version": "1.0"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled LinuxSubscriptionsDiscovery"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled LinuxSubscriptionsDiscovery"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings":
{
    "type": "structure",
    "members": {
        "SourceRegions": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
            "traits": {
                "smithy.api#documentation": "<p>The Regions in which to discover data for Linux
subscriptions.</p>",

```

```

        "smithy.api#required": {}
    },
    "OrganizationIntegration": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration",
        "traits": {
            "smithy.api#documentation": "<p>Details if you have enabled resource discovery across your
accounts in Organizations.</p>",
            "smithy.api#required": {}
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Lists the settings defined for discovering Linux
subscriptions.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstances": {
        "type": "operation",
        "input": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest"
        },
        "output": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse"
        },
        "errors": [
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
            },
            {
                "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
            }
        ],
        "traits": {
            "smithy.api#documentation": "<p>Lists
the running Amazon EC2 instances that were discovered with commercial Linux\n    subscriptions.</p>",
            "smithy.api#http": {
                "code": 200,
                "method": "POST",
                "uri": "/subscription/ListLinuxSubscriptionInstances"
            },
            "smithy.api#idempotent": {},
            "smithy.api#paginated": {
                "inputToken": "NextToken",
                "outputToken": "NextToken",

```

```

        "pageSize": "MaxResults",
        "items": "Instances"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesRequest": {
    "type": "structure",
    "members": {
        "Filters": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",
            "traits": {
                "smithy.api#documentation": "<p>An array of structures
that you can use to filter the results by your specified criteria. \n \tFor example, you can specify
<code>Region</code> in the <code>Name</code>, with the \n \t<code>contains</code> operator to list all
subscriptions that match a partial string in the \n \t<code>Value</code>, such as <code>us-west</code>.</p>\n
<p>For each filter, you can specify one of the following values for the <code>Name</code> key \n \tto
streamline results:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>AccountID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>AmiID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>DualSubscription</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceID</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>InstanceType</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>ProductCode</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Region</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Status</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>UsageOperation</code>\n
                </p>\n
            </li>\n
        </ul>\n
        <p>For each filter, you can use
one of the following <code>Operator</code> values to \n \tdefine the behavior of the filter:</p>\n
        <ul>\n
            <li>\n
                <p>\n
                    <code>contains</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>equals</code>\n
                </p>\n
            </li>\n
            <li>\n
                <p>\n
                    <code>Notequal</code>\n
                </p>\n
            </li>\n
        </ul>"
            }
        },
    },
    "MaxResults": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
        "traits": {
            "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
        }
    },
    "NextToken": {
        "target": "smithy.api#String",
        "traits": {
            "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
            "smithy.api#length": {
                "min": 1,
                "max": 16384
            }
        }
    }
}

```

```

    },
    "traits": {
      "smithy.api#documentation":
"NextToken length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionInstancesResponse": {
    "type": "structure",
    "members": {
      "Instances": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InstanceList",
        "traits": {
          "smithy.api#documentation": "<p>An array that contains instance objects.</p>"
        }
      },
      "NextToken": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the next request to retrieve \n\tadditional objects.</p>"
        }
      }
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptions": {
    "type": "operation",
    "input": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest"
    },
    "output": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse"
    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation":

```

"<p>Lists the Linux subscriptions that have been discovered. If you have linked your\n organization, the returned results will include data aggregated across your accounts in\n Organizations.</p>",

```
"smithy.api#http": {
  "code": 200,
  "method": "POST",
  "uri": "/subscription/ListLinuxSubscriptions"
},
"smithy.api#idempotent": {},
"smithy.api#paginated": {
  "inputToken": "NextToken",
  "outputToken": "NextToken",
  "pageSize": "MaxResults",
  "items": "Subscriptions"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsRequest": {
  "type": "structure",
  "members": {
    "Filters": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#FilterList",

      "traits": {
        "smithy.api#documentation": "<p>An array of structures that you can use to filter the results to those
that match one or\n more sets of key-value pairs that you specify. For example, you can filter by the name of\n
<code>Subscription</code> with an optional operator to see subscriptions that match,\n partially match, or don't
match a certain subscription's name.</p>\n <p>The valid names for this filter are:</p>\n <ul>\n
<li>\n <p>\n <code>Subscription</code>\n </p>\n </li>\n </ul>\n
<p>The valid Operators for this filter are:</p>\n <ul>\n <li>\n <p>\n
<code>contains</code>\n </p>\n </li>\n <li>\n <p>\n
<code>equals</code>\n </p>\n </li>\n <li>\n <p>\n
<code>NotEqual</code>\n </p>\n </li>\n </ul>"
      }
    },
    "MaxResults": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
      "traits": {
        "smithy.api#documentation": "<p>The maximum items to return in a request.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\tis the
nextToken from a previously truncated response.</p>",
        "smithy.api#length": {
          "min": 1,
          "max": 16384
        }
      }
    }
  }
}
```

```

    }
  }
},
"traits": {
  "smithy.api#documentation": "NextToken
length limit is half of ddb accepted limit.\nIncrease this limit if parameters in request increases."
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListLinuxSubscriptionsResponse": {
  "type": "structure",
  "members": {
    "Subscriptions": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList",
      "traits": {
        "smithy.api#documentation": "<p>An array that contains subscription objects.</p>"
      }
    },
    "NextToken": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The next token used for paginated responses. When this \n\tfield
isn't empty, there are additional elements that the service hasn't \n\tincluded in this request. Use this token with the
next request to retrieve \n\tadditional objects.</p>"
      }
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProviders": {
  "type": "operation",
  "input": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest"
  },
  "output": {
    "target":
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
    }
  ]
}

```

```

    }
  ],
  "traits": {
    "smithy.api#documentation":
"<p>List Bring Your Own License (BYOL) subscription registration resources for your account.</p>",
    "smithy.api#http": {
      "code": 200,
      "method": "POST",
      "uri": "/subscription/ListRegisteredSubscriptionProviders"
    },
    "smithy.api#idempotent": {},
    "smithy.api#paginated": {
      "inputToken": "NextToken",
      "outputToken": "NextToken",
      "pageSize": "MaxResults",
      "items": "RegisteredSubscriptionProviders"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListRegisteredSubscriptionProvidersRequest": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSources": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList",

"traits": {
  "smithy.api#documentation": "<p>To filter your results, specify which subscription providers to
return \n\t\t\tin the list.</p>"
    }
  },
  "MaxResults": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxInteger",
    "traits": {
      "smithy.api#documentation": "<p>The maximum items to return in a request.</p>",
      "smithy.api#range": {
        "min": 1,
        "max": 100
      }
    }
  },
  "NextToken": {
    "target": "smithy.api#String",
    "traits": {
      "smithy.api#documentation": "<p>A token to specify where to start paginating. This \n\t\tis the
nextToken from a previously truncated response.</p>"
    }
  }
},

```



```
"smithy.api#documentation": "<p>List the metadata tags that are assigned to the \n\t\t\tspecified Amazon Web Services resource.</p>",
    "smithy.api#http": {
      "method": "GET",
      "uri": "/tags/{resourceArn}"
    },
    "smithy.api#readonly": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceRequest": {
  "type": "structure",
  "members": {
    "resourceArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the resource for which to list metadata tags.</p>",
        "smithy.api#httpLabel": {},
        "smithy.api#required": {}
      }
    }
  },
  "traits": {
    "smithy.api#input": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ListTagsForResourceResponse": {
  "type": "structure",
  "members": {
    "tags": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Tags",
      "traits": {
        "smithy.api#documentation": "<p>The metadata tags for the requested resource.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Operator": {
  "type": "string",
  "traits": {
    "smithy.api#enum": [
      {
        "name": "EQUAL",
        "value": "Equal",
```

```

        "documentation": "Equal operator"
    },
    {
        "name": "NOT_EQUAL",
        "value": "NotEqual",
        "documentation": "Not equal operator"
    },
    {
        "name": "CONTAINS",
        "value": "Contains",
        "documentation": "Contains operator"
    }
],
"smithy.api#length": {
    "min": 1,
    "max": 20
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#OrganizationIntegration": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "Enabled",
                "value": "Enabled",
                "documentation": "Enabled OrganizationIntegration"
            },
            {
                "name": "Disabled",
                "value": "Disabled",
                "documentation": "Disabled OrganizationIntegration"
            }
        ]
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ProductCodeList": {
    "type": "list",
    "member": {
        "target": "smithy.api#String"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProvider": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderRequest"
    },

```



```

    "smithy.api#documentation": "<p>The metadata tags to assign to your registered Linux subscription
provider \n\t\t\tresource.</p>"
  }
},
"traits": {
  "smithy.api#input": {}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisterSubscriptionProviderResponse": {
  "type": "structure",
  "members": {
    "SubscriptionProviderSource": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
      "traits": {
        "smithy.api#documentation": "<p>The Linux subscription provider that you registered.</p>"
      }
    },
    "SubscriptionProviderArn": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of
the Linux subscription provider resource that you registered.</p>"
      }
    },
    "SubscriptionProviderStatus": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
      "traits": {
        "smithy.api#documentation": "<p>Indicates the status of the registration action for the Linux
subscription provider \n\t\t\t\tthat you requested.</p>"
      }
    }
  },
  "traits": {
    "smithy.api#output": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider": {
  "type": "structure",
  "members": {
    "SubscriptionProviderArn": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
      "traits": {
        "smithy.api#documentation": "<p>The
Amazon Resource Name (ARN) of the Linux subscription provider resource that you registered.</p>"
      }
    }
  },

```

```

"SubscriptionProviderSource": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource",
  "traits": {
    "smithy.api#documentation": "<p>A supported third-party Linux subscription provider. License Manager currently supports \n\t\t\tRed Hat subscriptions.</p>"
  }
},
"SecretArn": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn",
  "traits": {
    "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Secrets Manager secret that stores your registered Linux subscription provider \n\t\t\taccess token. For RHEL account subscriptions, this is the offline token.</p>"
  }
},
"SubscriptionProviderStatus": {
  "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus",
  "traits": {
    "smithy.api#documentation": "<p>Indicates the status of your registered Linux subscription provider access token \n\t\t\tfrom the last time License Manager retrieved subscription data. For RHEL account subscriptions, \n\t\t\tthis is the status of the offline token.</p>"
  }
},
"SubscriptionProviderStatusMessage": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>A detailed message that's associated with your BYOL subscription \n\t\t\tprovider token status.</p>"
  }
},
"LastSuccessfulDataRetrievalTime": {
  "target": "smithy.api#String",
  "traits": {
    "smithy.api#documentation": "<p>The timestamp from the last time that License Manager accessed third-party subscription data \n\t\t\tfor your account from your registered Linux subscription provider.</p>"
  }
},
"traits": {
  "smithy.api#documentation": "<p>A third-party provider for operating system (OS) platform software and license \n\t\t\tsubscriptions, such as Red Hat. When you register a third-party Linux subscription \n\t\t\tprovider, License Manager can get subscription data from the registered provider.</p>"
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProviderList": {
  "type": "list",
  "member": {

```

```

        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#RegisteredSubscriptionProvider"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException":
{
    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>Unable to find the requested Amazon Web Services resource.</p>",
        "smithy.api#error": "client"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#SecretArn": {
    "type": "string",
    "traits": {
        "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:secretsmanager:[a-z0-9-\\.]{0,63}:[a-z0-9-\\.]{0,63}:secret:[^/]{1,1023}$"
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#Status": {
    "type": "string",
    "traits": {
        "smithy.api#enum": [
            {
                "name": "InProgress",
                "value": "InProgress",
                "documentation": "InProgress status"
            },
            {
                "name": "Completed",
                "value": "Completed",
                "documentation": "Completed status"
            },
            {
                "name": "Successful",
                "value": "Successful",
                "documentation": "Successful status"
            },
            {
                "name": "Failed",
                "value": "Failed",
                "documentation": "Failed status"
            }
        ]
    }
}
]

```

```

    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringList": {
    "type": "list",
    "member": {
      "target": "smithy.api#String",
      "traits": {
        "smithy.api#length": {
          "min": 1,
          "max": 100
        }
      }
    },
    "traits": {
      "smithy.api#length": {
        "min": 1,
        "max": 100
      },
      "smithy.api#uniqueItems": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap": {
    "type": "map",
    "key": {
      "target": "smithy.api#String"
    },
    "value": {
      "target": "smithy.api#String"
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription": {
    "type": "structure",
    "members": {
      "Name": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The name of the subscription.</p>"
        }
      },
      "Type": {
        "target": "smithy.api#String",
        "traits": {
          "smithy.api#documentation": "<p>The type of subscription. The type can be subscription-included with Amazon EC2, Bring Your Own\n    Subscription model (BYOS), or from the Amazon Web Services Marketplace. Certain subscriptions may use licensing from the\n    Amazon Web Services Marketplace as well as OS licensing from Amazon EC2 or BYOS.</p>"
        }
      }
    }
  }

```



```

    },
    "InstanceCount": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#BoxLong",
      "traits": {
        "smithy.api#documentation": "<p>The total amount of running instances using this subscription.</p>"
      }
    },
    "traits": {
      "smithy.api#documentation": "<p>An object which details a discovered Linux subscription.</p>"
    },
    "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionList": {
      "type": "list",

      "member": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Subscription"
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn": {
        "type": "string",
        "traits": {
          "smithy.api#pattern": "^arn:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,63}:[a-z0-9-\\.]{1,510}/[a-z0-9-\\.]{1,510}$"
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource": {
        "type": "string",
        "traits": {
          "smithy.api#enum": [
            {
              "name": "RedHat",
              "value": "RedHat",
              "documentation": "RedHat subscription provider namespace"
            }
          ]
        }
      },
      "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSourceList": {
        "type":
"list",
        "member": {
          "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderSource"
        },
        "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderStatus": {
          "type": "string",
          "traits": {

```

```

    "smithy.api#enum": [
      {
        "name": "ACTIVE",
        "value": "ACTIVE",
        "documentation": "ACTIVE status"
      },
      {
        "name": "INVALID",
        "value": "INVALID",
        "documentation": "INVALID status"
      },
      {
        "name": "PENDING",
        "value": "PENDING",
        "documentation": "PENDING status"
      }
    ]
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList":
{
  "type": "list",
  "member": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#TagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerError"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ]
}

```



```

    "type": "map",
    "key":
  {
    "target": "smithy.api#String"
  },
  "value": {
    "target": "smithy.api#String"
  },
  "traits": {
    "smithy.api#length": {
      "min": 0,
      "max": 50
    },
    "smithy.api#sensitive": {}
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException": {
  "type": "structure",
  "members": {
    "message": {
      "target": "smithy.api#String"
    }
  },
  "traits": {
    "smithy.api#documentation": "<p>The request was denied due to request throttling.</p>",
    "smithy.api#error": "client"
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResource": {
  "type": "operation",
  "input": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest"
  },
  "output": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse"
  },
  "errors": [
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
    },
    {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ResourceNotFoundException"
    }
  ],
  "traits": {
    "smithy.api#documentation": "<p>Remove one or more metadata tag from the specified Amazon Web Services resource.</p>",
    "smithy.api#http": {

```

```

        "method": "DELETE",
        "uri": "/tags/{resourceArn}"
    },
    "smithy.api#idempotent": {}
}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceRequest": {
    "type": "structure",

    "members": {
        "resourceArn": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#SubscriptionProviderArn",
            "traits": {
                "smithy.api#documentation": "<p>The Amazon Resource Name (ARN) of the Amazon Web Services
resource to remove the metadata tags from.</p>",
                "smithy.api#httpLabel": {},
                "smithy.api#required": {}
            }
        },
        "tagKeys": {
            "target": "com.amazonaws.licensemanagerlinuxsubscriptions#TagKeyList",
            "traits": {
                "smithy.api#documentation": "<p>A list of metadata tag keys to remove from the requested
\n\t\t\tresource.</p>",
                "smithy.api#httpQuery": "tagKeys",
                "smithy.api#required": {}
            }
        }
    },
    "traits": {
        "smithy.api#input": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UntagResourceResponse": {
    "type": "structure",
    "members": {},
    "traits": {
        "smithy.api#output": {}
    }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettings": {
    "type": "operation",
    "input": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest"
    },
    "output": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse"
    }
}

```

```

    },
    "errors": [
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#InternalServerErrorException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ThrottlingException"
      },
      {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException"
      }
    ],
    "traits": {
      "smithy.api#documentation": "<p>Updates the service settings for Linux subscriptions.</p>",
      "smithy.api#http": {
        "code": 200,
        "method": "POST",
        "uri": "/subscription/UpdateServiceSettings"
      },
      "smithy.api#idempotent": {}
    }
  },
  "com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsRequest": {
    "type": "structure",
    "members": {
      "LinuxSubscriptionsDiscovery": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
        "traits": {
          "smithy.api#documentation": "<p>Describes if the discovery of Linux subscriptions is enabled.</p>",
          "smithy.api#required": {}
        }
      },
      "LinuxSubscriptionsDiscoverySettings": {
        "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
        "traits": {
          "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated from.</p>",
          "smithy.api#required": {}
        }
      },
      "AllowUpdate": {
        "target": "smithy.api#Boolean",
        "traits": {
          "smithy.api#documentation": "<p>Describes if updates are allowed to the service settings for Linux subscriptions. If you\n    allow updates, you can aggregate Linux subscription data in more than one home

```

```

Region.</p>"
    }
}

}
},
"com.amazonaws.licensemanagerlinuxsubscriptions#UpdateServiceSettingsResponse": {
  "type": "structure",
  "members": {
    "LinuxSubscriptionsDiscovery": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscovery",
      "traits": {
        "smithy.api#documentation": "<p>Lists if discovery has been enabled for Linux subscriptions.</p>"
      }
    },
    "LinuxSubscriptionsDiscoverySettings": {
      "target": "com.amazonaws.licensemanagerlinuxsubscriptions#LinuxSubscriptionsDiscoverySettings",
      "traits": {
        "smithy.api#documentation": "<p>The settings defined for Linux subscriptions discovery. The
settings include if Organizations\n    integration has been enabled, and which Regions data will be aggregated
from.</p>"
      }
    }
  },
  "Status": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#Status",
    "traits": {
      "smithy.api#documentation": "<p>Indicates the status of Linux subscriptions settings being
applied.</p>"
    }
  },
  "StatusMessage": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringMap",
    "traits": {
      "smithy.api#documentation": "<p>A message which details the Linux subscriptions service settings
current status.</p>"
    }
  },
  "HomeRegions": {
    "target": "com.amazonaws.licensemanagerlinuxsubscriptions#StringList",
    "traits": {
      "smithy.api#documentation": "<p>The Region in which License Manager displays the aggregated
data for Linux\n    subscriptions.</p>"
    }
  }
},
"com.amazonaws.licensemanagerlinuxsubscriptions#ValidationException": {

```

```

    "type": "structure",
    "members": {
        "message": {
            "target": "smithy.api#String"
        }
    },
    "traits": {
        "smithy.api#documentation": "<p>The provided input is not valid. Try your request again.</p>",
        "smithy.api#error": "client"
    }
}
}
}
}

```

Apache License
 Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

AssociateLicense

Initialize stack step

spanInitializeStart

InterceptExecution

RegisterServiceMetadata

legacyEndpointContextSetter

SetLogger

OperationInputValidation

spanInitializeEnd

Serialize stack step

spanBuildRequestStart

setOperationInput

ResolveEndpoint

InterceptBeforeSerialization

OperationSerializer

InterceptAfterSerialization

Build stack step

ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit
ListLicenses
Initialize stack step
spanInitializeStart
InterceptExecution
RegisterServiceMetadata
legacyEndpointContextSetter
SetLogger
spanInitializeEnd
Serialize stack step
spanBuildRequestStart
setOperationInput
ResolveEndpoint
InterceptBeforeSerialization

OperationSerializer
InterceptAfterSerialization
Build stack step
ClientRequestID
ComputeContentLength
SetCredentialSourceMiddleware
UserAgent
AddTimeOffsetMiddleware
RecursionDetection
spanBuildRequestEnd
Finalize stack step
spanRetryLoop
InterceptBeforeRetryLoop
Retry
InterceptAttempt
RetryMetricsHeader
ResolveAuthScheme
GetIdentity
ResolveEndpointV2
disableHTTPS
ComputePayloadHash
setLegacyContextSigningOptions
InterceptBeforeSigning
Signing
InterceptAfterSigning
Deserialize stack step
AddRawResponseToMetadata
ErrorCloseResponseBody
CloseResponseBody
ResponseErrorWrapper
RequestIDRetriever
InterceptAfterDeserialization
OperationDeserializer
InterceptBeforeDeserialization
AddTimeOffsetMiddleware
RecordResponseTiming
RequestResponseLogger
InterceptTransmit

1.76 bash 5.2.37-r0

1.76.1 Available under license :

Shellfloat is copyright (c) 2020 by Michael Wood.

=====

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work
released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code.

And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed

this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying,

distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source

includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met.

This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your

work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no

further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied

by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this

License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or

author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation

by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept

this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if

the predecessor
has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner

consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.

If the

Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY

APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM

IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<https://www.gnu.org/licenses/why-not-lgpl.html>>.

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to

assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We

have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The

Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve

the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five),

unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the

Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See

<http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed

to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies
of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

@c The GNU Free Documentation License.

@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,
@c hence no sectioning command or @node.

@display

Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

@end display

@enumerate 0

@item

PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document @dfn{free} in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative
works of the document
must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that
contains a notice placed by the copyright holder saying it can be
distributed under the terms of this License. Such a notice grants a
world-wide, royalty-free license, unlimited in duration, to use that
work under the conditions stated herein. The ``Document'', below,
refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent" is called ``Opaque".

Examples of suitable formats for Transparent copies include plain `@sc{ascii}` without markup, Texinfo input format, `LaTeX{} input format`, `@acronym{SGML}` or `@acronym{XML}` using a publicly available `@acronym{DTD}`, and standard-conforming simple `@acronym{HTML}`, PostScript or `@acronym{PDF}` designed for human modification. Examples of transparent image formats include `@acronym{PNG}`, `@acronym{XCF}` and `@acronym{JPG}`. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, `@acronym{SGML}` or `@acronym{XML}` for which the `@acronym{DTD}` and/or processing tools are not generally available, and the machine-generated `@acronym{HTML}`, PostScript or `@acronym{PDF}` produced by some word processors for output purposes only.

The ``Title Page'' means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page'' means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher'' means any person or entity that distributes copies of the Document to the public.

A section ``Entitled XYZ'' means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements'', ``Dedications'', ``Endorsements'', or ``History''.) To ``Preserve the Title'' of such a section when you modify the Document means that it remains a section ``Entitled XYZ'' according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever

to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least

four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements''. Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements'' or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,

you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements'', provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History'' in the various original documents, forming one section Entitled

``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract

a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the

original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new

versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [@uref{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ``or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item
RELICENSING

``Massive Multiauthor Collaboration Site" (or ``MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A ``Massive Multiauthor Collaboration" (or ``MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

``CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the

MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ } Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with the Front-Cover Texts being @var{list}, and with the Back-Cover Texts being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to
freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as

changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without

permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally

available free

programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated

place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the

User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or

limiting liability differently from the terms of sections 15 and 16 of this License; or

- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights

granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to
do, use the GNU Lesser General
Public License instead of this License. But first, please read
<<http://www.gnu.org/philosophy/why-not-lgpl.html>>.
Unless otherwise stated, all files in this directory are Copyright (C)
1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,
2004,2005,2006,2007,2008,2009,2010,2011
Free Software Foundation, Inc.

See the file COPYING in the bash distribution root directory for copying
and usage restrictions.

The file ifs-posix.tests is Copyright (C) 2005 Glen Fowler.

1.77 x-sys 0.37.0

1.77.1 Available under license :

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.78 zlib 1.3.1-r2

1.78.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it

freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.79 iputils 20240905-r0

1.79.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it

if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in

themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component

itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals

of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by

the Free Software Foundation; either version 2 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.80 cJSON 1.7.19-r0

1.80.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit
(<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young (eyay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.80.2 Available under license :

Apache License 2.0

Contributors

=====

Original Author:

- [Dave Gamble](<https://github.com/DaveGamble>)

Current Maintainer:

- [Max Bruckner](<https://github.com/FSMaxB>)

- [Alan Wang](<https://github.com/Alanscut>)

Contributors:

- * [Ajay Bhargav](<https://github.com/ajaybhargav>)
- * [Alexander Vasiljev](<https://github.com/AlexanderVasiljev>)
- * [Alper Akcan](<https://github.com/alperakcan>)
- * [Andrew Tang](<https://github.com/singku>)
- * [Andy](<https://github.com/mlh0101>)
- * [Anton Sergeev](<https://github.com/anton-sergeev>)
- * [Benbuck Nason](<https://github.com/bnason-nf>)
- * [Bernt Johan Damslara](<https://github.com/bjda>)
- * [Bob Kocisko](<https://github.com/bobkocisko>)
- * [Christian Schulze](<https://github.com/ChristianSch>)
- * [Casperinous](<https://github.com/Casperinous>)
- * [ChenYuan](<https://github.com/zjuchenyuan>)
- * [Debora Grosse](<https://github.com/DeboraG>)
- * [dieyushi](<https://github.com/dieyushi>)
- * [Dngwn Hung ()](<https://github.com/DongwenHuang>)
- * [Dominik](<https://github.com/DL6ER>)
- * [Donough Liu](<https://github.com/ldm0>)
- *
- [Erez Oxman](<https://github.com/erez-o>)
- * Eswar Yaganti
- * [Evan Todd](<https://github.com/etodd>)
- * [Fabrice Fontaine](<https://github.com/ffontaine>)
- * Ian Mobley
- * Irwan Djadjadi
- * [hopper-vul](<https://github.com/hopper-vul>)
- * [HuKeping](<https://github.com/HuKeping>)
- * [IvanVoid](<https://github.com/npi3pak>)
- * [Jakub Wilk](<https://github.com/jwilk>)
- * [Jiri Zouhar](<https://github.com/loigu>)

- * [Jonathan Fether](https://github.com/jfether)
- * [Joshua Arulsamy](https://github.com/jarulsamy)
- * [Julian Ste](https://github.com/julian-st)
- * [Julin Vsquez](https://github.com/juvasquezg)
- * [Junbo Zheng](https://github.com/Junbo-Zheng)
- * [Kevin Branigan](https://github.com/kbranigan)
- * [Kevin Sapper](https://github.com/sappo)
- * [Kyle Chisholm](https://github.com/ChisholmKyle)
- * [Linus Wallgren](https://github.com/ecksun)
- * [Luo Jin](https://github.com/Up-wind)
- * [Max](https://github.com/maebex)
- * [MaxBrandtner](https://github.com/MaxBrandtner)
- * [Mateusz Szafoni](https://github.com/raiden00pl)
- *
- Mike Pontillo
- * [miaoerduo](https://github.com/miaoerduo)
- * [mohawk2](https://github.com/mohawk2)
- * [Mike Jerris](https://github.com/mjerris)
- * [Mike Robinson](https://github.com/mhrobinson)
- * [Moorthy](https://github.com/moorthy-bs)
- * [myd7349](https://github.com/myd7349)
- * [NancyLi1013](https://github.com/NancyLi1013)
- * [Orri](https://github.com/sbvoxel)
- * Paulo Antonio Alvarez
- * [Pawe Malowany](https://github.com/PawelMalowany)
- * [Pawel Winogrodzki](https://github.com/PawelWMS)
- * [prefetchnta](https://github.com/prefetchnta)
- * [Rafael Leal Dias](https://github.com/rafaeldias)
- * [Randy](https://github.com/randy408)
- * [raiden00pl](https://github.com/raiden00pl)
- * [Robin Mallinson](https://github.com/rmallins)
- * [Rod Vagg](https://github.com/rvagg)
- * [Roland Meertens](https://github.com/rmeertens)
- * [Romain Porte](https://github.com/MicroJoe)
- * [SANJEEV BA](https://github.com/basanjeev)
- * [Sang-Heon Jeon](https://github.com/Intuition)
- * [Sayan Bandyopadhyay](https://github.com/saynb)
- *
- [Simon Sobisch](https://github.com/GitMensch)
- * [Simon Ricaldone](https://github.com/simon-p-r)
- * [Stoian Ivanov](https://github.com/sdrsdr)
- * [SuperH-0630](https://github.com/SuperH-0630)
- * [Square789](https://github.com/Square789)
- * [Stephan Gatzka](https://github.com/gatzka)
- * [Tony Langhammer](https://github.com/BigBrainAFK)
- * [Vemake](https://github.com/vemakereporter)
- * [vwvw](https://github.com/vwvw)
- * [warmsocks](https://github.com/warmsocks)

- * [Wei Tan](https://github.com/tan-wei)
- * [Weston Schmidt](https://github.com/schmidtw)
- * [xiaomianhehe](https://github.com/xiaomianhehe)
- * [yangfl](https://github.com/yangfl)
- * [yuta-oxo](https://github.com/yuta-oxo)
- * [Zach Hindes](https://github.com/zhindes)
- * [Zhao Zhixu](https://github.com/zhaozhixu)
- * [10km](https://github.com/10km)

And probably more people on [SourceForge](https://sourceforge.net/p/cjson/bugs/search/?q=status%3Aclosed-rejected+or+status%3Aclosed-out-of-date+or+status%3Awont-fix+or+status%3Aclosed-fixed+or+status%3Aclosed&page=0)

Also

thanks to all the people who reported bugs and suggested new features.

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF

OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) <year> 2007-14 Mike Karlesky, Mark VanderVoord, Greg Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2025 Cisco Systems, Inc. All rights reserved.